NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

ANALYSIS OF CONSTANT PHASE CONTOURS OF
EVAPORATION DUCT MODE FUNCTIONS
FOR WAVEGUIDE MODE PROPAGATION

by

Jen-Peng Che

September 1992

Thesis Advisor: Hung-Mou Lee

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION UNCLASSFIED	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Naval Postgraduate School 6b. OFFICE SYMBOL (*If applicable*) EC	7a. NAME OF MONITORING ORGANIZATION Naval Postgraduate School
6c. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000	7b. ADDRESS (*City, State, and ZIP Code*) Monterey, CA 93943-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION 8b. OFFICE SYMBOL (*If applicable*)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
8c. ADDRESS (*City, State, and ZIP Code*)	10. SOURCE OF FUNDING NUMBERS Program Element No. / Project No. / Task No / Work Unit Accession Number

11. TITLE (*Include Security Classification*)
ANALYSIS OF CONSTANT PHASE CONTOURS OF EVAPORATION DUCT MODE FUNCTIONS FOR WAVEGUIDE MODE PROPAGATION

12. PERSONAL AUTHOR(S) Jen-Peng Che

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (*year, month, day*)	15. PAGE COUNT
Master's Thesis	From To	September 1992	101

16. SUPPLEMENTARY NOTATION
The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

17. COSATI CODES			18. SUBJECT TERMS (*continue on reverse if necessary and identify by block number*)
FIELD	GROUP	SUBGROUP	Evaporation Duct, Waveguide Mode Propagation

19. ABSTRACT (*continue on reverse if necessary and identify by block number*)

The M-Layer program tracks the constant phase lines $\mathrm{Im}\{D(q)\} = 0$ and looks for their intersections with the lines $\mathrm{Re}\{D(q)\} = 0$ for the locations of the zeros of the mode function $D(q)$. These two types of constant phase lines are tracked and plotted over a search region which contains modes having a range attenuation rate of no more than 5 dB per km. Several new parameters for use in mode search are deduced from the results and some old ones are verified. Future studies in waveguide mode propagation theory pertaining to atmospheric ducts may benefit from this work. An improved mode search strategy is also proposed.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION	
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS REPORT ☐ DTIC USERS	UNCLASSFIED	
22a. NAME OF RESPONSIBLE INDIVIDUAL Hung-Mou Lee	22b. TELEPHONE (*Include Area code*) (408)646-2846	22c. OFFICE SYMBOL EC/Lh

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

Approved for public release; distribution is unlimited.

ANALYSIS OF CONSTANT PHASE CONTOURS OF EVAPORATION DUCT MODE FUNCTIONS FOR WAVEGUIDE MODE PROPAGATION

by

Jen-Peng Che
Lieutenant Commander, Taiwan Navy
B.S., Chinese Naval Academy, 1980

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

from the

NAVAL POSTGRADUATE SCHOOL
September 1992

Author: _____
Jen-Peng Che

Approved by: _____
Hung-Mou Lee, Thesis Advisor

_____
Lawrence J. Ziomek, Second Reader

_____
Michael A. Morgan, Chairman
Department of Electrical and Computer Engineering

ABSTRACT

The M-Layer program tracks the constant phase lines Im{D(q)} = 0 and looks for their intersections with the lines Re{D(q)} = 0 for the locations of the zeros of the mode function D(q). These two types of constant phase lines are tracked and plotted over a search region which contains modes having a range attenuation rate of no more than 5 dB per km. Several new parameters for use in mode search are deduced from the results and some old ones are verified. Future studies in waveguide mode propagation theory pertaining to atmospheric ducts may benefit from this work. An improved mode search strategy is also proposed.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my advisor Dr. Hung-Mou Lee for his professional guidance, assistance and support throughout the thesis preparation period. I thank my second reader Dr. Lawrence J. Ziomek for his careful review and constructive suggestions to the thesis work.

Special thanks to my wife Mei-Ying Huang and my son Yu-Heng Che for their love and unending support which allowed me to successfully complete my graduate education.

I. INTRODUCTION

The M-Layer program developed by NOSC (Naval Ocean System Center) was documented by Yeoh [Ref. 1] at NPS. It was later extensively revised by Lee and Han [Ref. 2] for improved accuracy and speed. The new FORTRAN code is now identified as the NPS version [Ref. 3] under the auspices of NRaD (Naval Command, Control and Ocean Surveillance Center, RDT&E Division), the current name for NOSC. In this version, the logical structures and numerical algorithms for following the constant phase lines of the mode function and for computing the mode locations were almost completely rewritten. The overall plan to first partition the search region into rectangular areas and then circulate around these rectangles to search for the modes, i.e., the mode search protocol, was left unchanged. Lee and Han [Ref. 2] recommended that a more direct mode search protocol should be devised to locate the modes more expediently. If possible, such an approach should locate the modes in the order of ascending range attenuation rates.

To design a more efficient mode search protocol, a better understanding of the mode function beyond the known locations of the modes is required. In this thesis, the analytical property of the mode function is investigated. Programs are written to track and plot the constant phase lines along which the mode function is either real or purely imaginary. From the results, a new strategy to locate the modes is recommended.

In the remainder of this chapter, the background theory and some of the notations used in this thesis are introduced. The basic theory presented in Sections A and B follows Ref. 3 closely. The results of this research are presented in Chapter II. In Chapter III, a new mode search protocol is proposed based on the findings of this thesis. The relevant parameters for use in this new approach are also discussed.

A. THE WAVEGUIDE MODE THEORY OF PROPAGATION

Trapping of electromagnetic (EM) waves in the modes supported by a duct is the dominating factor in over-the-horizon propagation. The computer program M-Layer searches for these modes and computes the electric field from the Hertz vector. Assume a vertical electric dipole

$$\vec{J} = 4\pi\hat{z}\delta(x)\delta(y)\delta(z-z_T)$$

at a height $z=z_T$ above the surface of the earth. Following Freehafer [Ref. 4], the Hertz vector of the EM fields can be written, in the cylindrical coordinates (ρ, ϕ, z), as a sum of contributions from individual modes [Ref. 1]:

$$\Pi(\rho,z) = -\pi j \sum_m H_0^{(2)}(\beta_m \rho) g_m(z_T) g_m(z) \qquad (1)$$

where β_m is the wavenumber of the m-th mode and is independent of the coordinates; z_T is the height of the transmitter above the ground, $H_0^{(2)}$ is the Hankel function of the second kind, which represents an outgoing wave in the radial direction when the time

dependence $e^{j\omega t}$ is adopted, and g_m is the height-gain function of the m-th mode, normalized so that the integral of its square over all heights equals unity when either an electric or a magnetic dipole source is used. The height-gain function satisfies the equation

$$\left(\frac{\partial^2}{\partial z^2}+k^2m^2(z)-\beta_m^2\right)g_m(z)=0 \tag{2}$$

where k is the free-space wavenumber, m(z) is the modified index of refraction, and $m^2(z)$ is approximated with a continuous, piecewise linear profile having I layers:

$$m^2(z)=m_i^2+\alpha_i(z-z_i) \qquad z_i \le z \le z_{i+1}, \quad 1 \le i \le I. \tag{3}$$

In practice, m(z) deviates only slightly from unity in the troposphere. The modified refractivity M(z) = [m(z)-1]$\times 10^6$, as shown in Fig. 1, is commonly used. The slope of M(z) is approximately $\alpha_i \times 10^6/2$.

In the i-th layer, the height-gain function is given by

$$g_m(z)=B_i(\beta_m)[C_i(\beta_m)k_1(q_{mi})+k_2(q_{mi})] \qquad z_i \le z \le z_{i+1} \tag{4}$$

where q_{mi} is a dimensionless linear function of height z with k, β_m, m_i, and α_i as parameters:

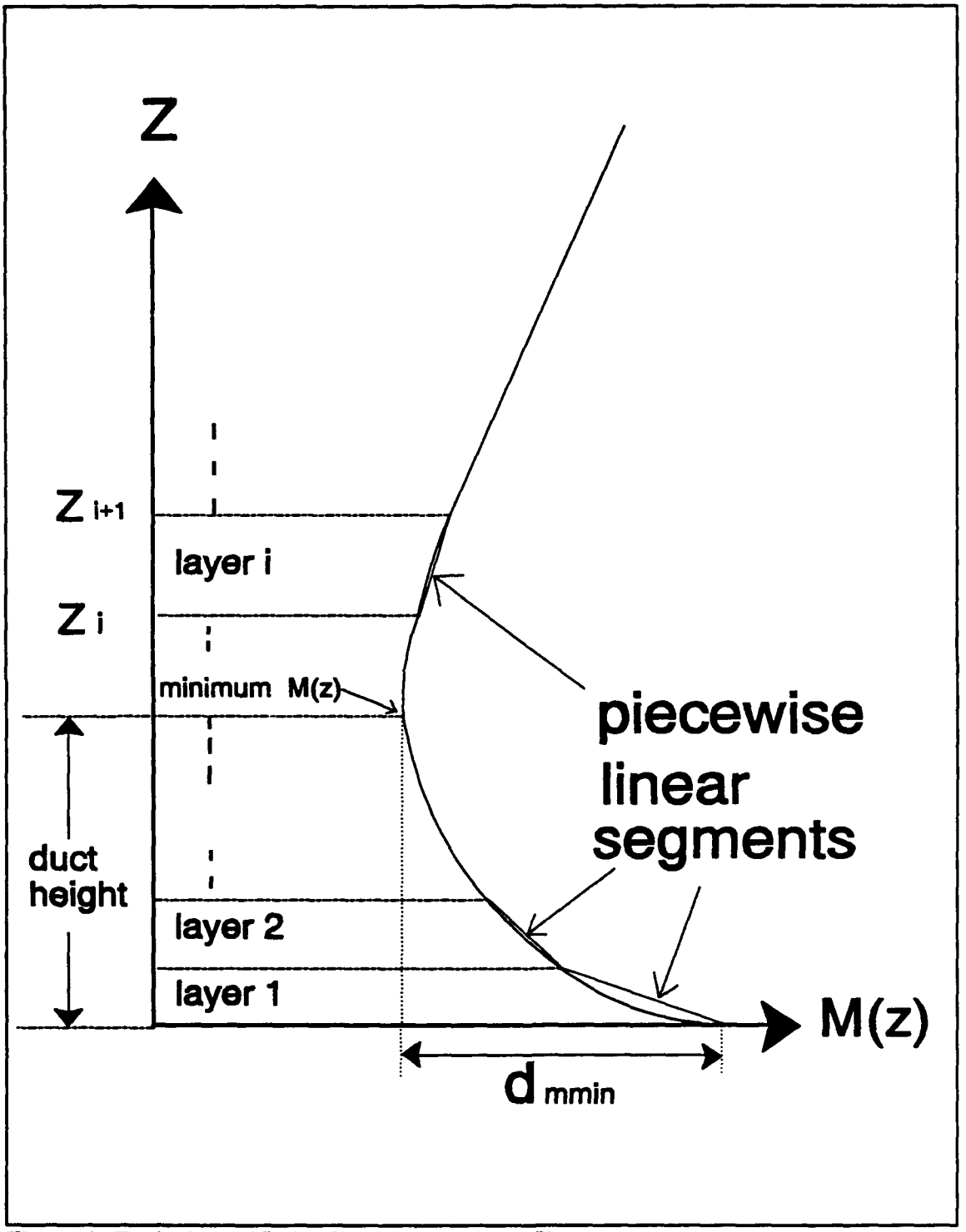


Figure 1. Typical modified refractivity M(z) profile.

$$q_{mi} = \sqrt[3]{\left(\frac{k}{\alpha_i}\right)^2}\left(m_i^2 + \alpha_i(z - z_i) - \frac{\beta_m^2}{k^2}\right) \tag{5}$$

The functions $k_1(q_{mi})$ and $k_2(q_{mi})$ are given in terms of the Airy function Ai:

$$k_1(q_{mi}) = -j2(12)^{1/6} Ai(-q_{mi} e^{j2\pi/3}) \tag{6}$$

and

$$k_2(q_{mi}) = 2(12)^{1/6} Ai(-q_{mi}). \tag{7}$$

For $z > z_{I+1}$, $m^2(z)$ is extended continuously upward at a constant slope commensurate with the effective radius of the earth. This slope equals 2.36×10^{-7} (m^{-1}) for the four-thirds effective earth radius model. Thus, the height-gain function g_m is again given by Eqs. (4) through (7) for $z > z_{I+1}$, with the constant C_{I+1} set to $e^{j4\pi/3}$ to represent an upward going wave as z becomes large. Beneath the "flattened" earth surface where $z < z_1 = 0$, $m^2(z)$ is deduced from the dielectric constant and the conductivity of sea water, which are set to 80.8869 and 4.64 (Si/m) respectively. Here the Hertz vector is represented by a downward propagating plane wave in this region.

The conditions that the tangential components of the electric and magnetic fields are continuous across the layer boundaries determine the coefficients C_i. Starting from the top layer down, or "integration-down" [Ref. 1], every C_i is uniquely determined by C_{i+1} at $z = z_{i+1}$. Thus, a set of all C_i coefficients is determined without considering the boundary conditions at $z = z_1$. On the other hand, starting from the lowest boundary at

$z = z_1$ up to $z = z_I$, or "integration-up", a second set of C_i coefficients is determined without applying the boundary conditions at $z = z_{I+1}$. That these two sets of C_i coefficients are identical is a manifestation that β_m is the wavenumber of a mode. Mathematically, this consistency condition, sometimes called the guidance condition [Ref. 5], can be expressed as the vanishing of a determinant D called the mode function [Ref. 6]. The elements of this determinant consist of linear combinations of $k_1(q_{mi})$ and $k_2(q_{mi})$ and their derivatives with respect to height at the boundaries. The M-Layer program searches for the zeros of the mode function to find the wavenumbers. Once a wavenumber is obtained, the height-gain function of the corresponding mode can be computed. Note that the normalization condition on g_m and the boundary conditions, together with the C_i coefficients, determine the B_i coefficients to within an overall sign. *This sign need not be resolved* because only products in the form of $g_m(z)g_m(z_T)$ from each mode contribute to the Hertz vector.

Since the mode function D depends on the wavenumber β_m only through the values of q_{mi} at the layer boundaries, it is more convenient to consider D as a function of the variable q given by

$$q = \sqrt[3]{\left(\frac{k}{\alpha_1}\right)^2\left(m_1^2 - \frac{\beta^2}{k^2}\right)}, \tag{8}$$

and to search for the zeros of D(q) in the complex q plane. The m-th zero is designated as q_m and is called a q-eigenvalue and β_m is then deduced from q_m by inverting Eq. (8)

for β. In the NPS version, q_m is ordered in ascending attenuation rate of the mode, which is approximately proportional to the imaginary part of β_m.

B. MODE SEARCH PRINCIPLE

M-Layer searches for modes which have attenuation rates below a specified value. At ground ranges more than several wavelengths away, this attenuation rate is proportional to the imaginary part of the wavenumber β_m as can be seen from Eq. (1). In the upper complex q plane, for values of q such that

$$\left| q\left(\frac{k}{\alpha_1}\right)^{-2/3} \right| = \left| m_1^2 - \frac{\beta^2}{k^2} \right| << 1, \tag{9}$$

$m_1 - \beta/k$ is approximately proportional to q because the absolute value of m_1 and, hence, that of β/k, are both nearly unity. Thus, the limit on attenuation rate imposed on the imaginary part of β can be translated approximately into an upper bound[1] on the imaginary part of q which defines a strip in the complex q plane called the search region. This region is covered with layers of identical square meshes whose sides are parallel to the imaginary q axis and have a length equivalent to an attenuation rate of 1/32 dB[2] per

---

[1] Since the absorption is small in air, the modified index of refraction m(z), and hence α_i, are considered as real quantities in the following discussions. In the actual FORTRAN code, they are declared as complex variables.

[2] This is the default value of the NPS version which can be adjusted through editing the variable "dmesh" in an ASCII input file.

kilometer. The lower edges of the meshes in the lowest layer sit along the real q axis[3]. The meshes in the layer just below the top one contain the upper bound of the search region, with the top layer providing some allowance for numerical inaccuracy. The program tests each mesh square for the presence of zeros of the mode function D(q).

To search through all the meshes, the program first divides the mesh covering the search region into "contour rectangles" with equally spaced vertical lines parallel to the imaginary q axis. These vertical lines contain the edges of stacked square meshes and are separated by a distance 160 times the side of a mesh. The search commences at the top left corner and moves counterclockwise around each "contour rectangle" and begins with the one whose left edge is defined by

$$Re\left[q\left(\frac{k}{\alpha_1}\right)^{-2/3}\right] = Re\left(m_1^2 - m_{min}^2\right) \approx 2\times10^{-6} Re\left(M_1 - M_{min}\right) = 2\times10^{-6} d_{mmin}, \qquad (10)$$

where m_{min} is the minimum value of the modified index of refraction profile and d_{mmin} is shown in Fig. 1. The justification for this choice to begin the mode search is as follows: From Eq. (5) and optics, for a wave to propagate freely in space, the dispersion relation of the Maxwell equations requires that $k^2m^2(z) > \beta^2_m$, assuming that both quantities are real. Thus, the smallest β^2_m to support a trapped wave will occur when

[3] The NOSC version extends the lower edge slightly below the real q-axis. This causes problems in some situations [Ref. 2].

β^2_m just exceeds the minimum of $m^2(z)^4$. It is not surprising that an argument based on optics works within a waveguide mode formulation which is low frequency in principle. Lee [Ref. 7] has demonstrated that the earth-flattening approximation actually links Mie's low frequency oriented spherical harmonics to Fock's high frequency diffraction theory through the uniform asymptotics. The mere introduction of an unlimited ground range into the Maxwell differential equations incorporates the global feature of the radius of the earth into the local equations.

After the search over the initial rectangle is completed, the program goes on to search the neighboring rectangle to the left. If a specified number[5] of consecutive rectangles of decreasing real q values have been searched without turning up any zero of D(q), the program changes direction and starts to search the rectangles to the right of the initial "contour rectangle" one by one, with increasing real part of q. After failing consecutively to locate any q-eigenvalue again over the specified number of rectangles, the program assumes that no more zeros of D(q) exists in the search region. The mode search is considered complete and the procedure is terminated. If the array for storing the q-eigenvalues is filled up before the search is completed, the search is terminated with an error message.

---

[4] Note that assuming the same real part, an imaginary component of β_m reduces the real part of β^2_m which may enable the wave to propagate in the z-direction.

[5] This number of consecutive rectangles is an adjustable input variable "nstop" in the NPS version. The default value is 2.

The search for zeros of D(q) makes use of the fact that a real valued function changes sign when it crosses a simple zero. Since a zero of the complex valued function D(q) is where both its real part and imaginary part vanish, a necessary condition for a point q_m to be a zero is that it is the intersection of two curves defined by $Im\{D(q)\} = 0$ and $Re\{D(q)\} = 0$. M-Layer moves along each side of a "contour rectangle" while searching for a sign change in $Im\{D(q)\}$ across an edge of a mesh bordering the side of this rectangle to determine that a line of $Im\{D(q)\} = 0$ has been encountered. The search then follows this line into the meshes within the "contour rectangle", checking each mesh to see if a curve $Re\{D(q)\} = 0$ enters the mesh being inspected. All these steps make use of the assumption that the zeros of D(q) are simple. Once both the curve $Im\{D(q)\} = 0$ and the curve $Re\{D(q)\} = 0$ are found to be present within a mesh, the locations of their possible intersections are estimated.

The rule for sign change becomes inconclusive if a zero of $Im\{D(q)\} = 0$ or if $Re\{D(q)\} = 0$ happens to fall on a corner of a mesh, and a remedy is required. In the NPS version, whenever the real part or the imaginary part of D(q) vanishes on a corner of a mesh, the phase angle of D(q) is rotated by 2^{-52} radians. This maneuver effectively shifts q by a small amount to resolve the sign ambiguity.

To estimate the locations of zeros of D(q) within a mesh, D(q) is assumed to be well approximated in this mesh by its four-term Taylor series expansion. The unshifted values of D(q) at the mesh corners determine this cubic polynomial uniquely. Cardan's formulas [Ref. 8] are used to locate the zeros of this polynomial before retaining only those lying within the mesh.

II. MODE FUNCTION IN THE COMPLEX q_{11}/t_1 PLANE

A. MODE LOCATIONS

M-Layer searches for the zeros of the mode function D(q) in the complex q plane by tracking the constant phase lines of D(q) along which the mode function is either real or purely imaginary. The zeros found are called the q-eigenvalues as defined by Eq. (8) and are denoted as q_m. These eigenvalues are saved in an ASCII file and are utilized later for height-gain function computations. In order to design a strategy for locating these zeros, the mode locations are plotted.

It is clear from Eq. (8) that the variable q varies with α_1, the slope of $m^2(z)$ in the lowest (first) layer. Since α_1 depends on how the continuous, piecewise linear approximation to $m^2(z)$ is made, while both m_1 and β_m are, in principle, dependent only on the actual profile, q will vary strongly with α_1 and fail to provide information on the wavenumber of the mode directly. A more suitable variable to use is, from Eq. (8):

$$q\left(\frac{k}{\alpha_1}\right)^{-2/3} = m_1^2 - \frac{\beta^2}{k^2}. \tag{11}$$

Since q is given the variable name of q_{11} and $(k/\alpha_1)^{2/3}$ is given the variable name of t_1 in the M-Layer FORTRAN code, this variable is identified as q_{11}/t_1 throughout this thesis. Since m_1 is real and both it and β/k deviate from unity only slightly for all cases

investigated, q_{11}/t_1 is approximately $2(m_1 - \beta/k)$. Thus, its imaginary part is directly proportional to $-\beta$, the attenuation rate of the mode. Furthermore, q_{11}/t_1 does not depend explicitly on α_1. Removing the t_1 factor from q_{11} makes it possible to compare results from different evaporation duct profiles meaningfully.

Plots of the q-eigenvalues as individual points in the complex q_{11}/t_1 plane are included in Appendix A. For the 20 meter duct, which will be used as the representative case for discussions in this thesis, a line linking all the eigenvalues in the order of increasing attenuation rate is shown in Fig. 2. Because of the long excursion between

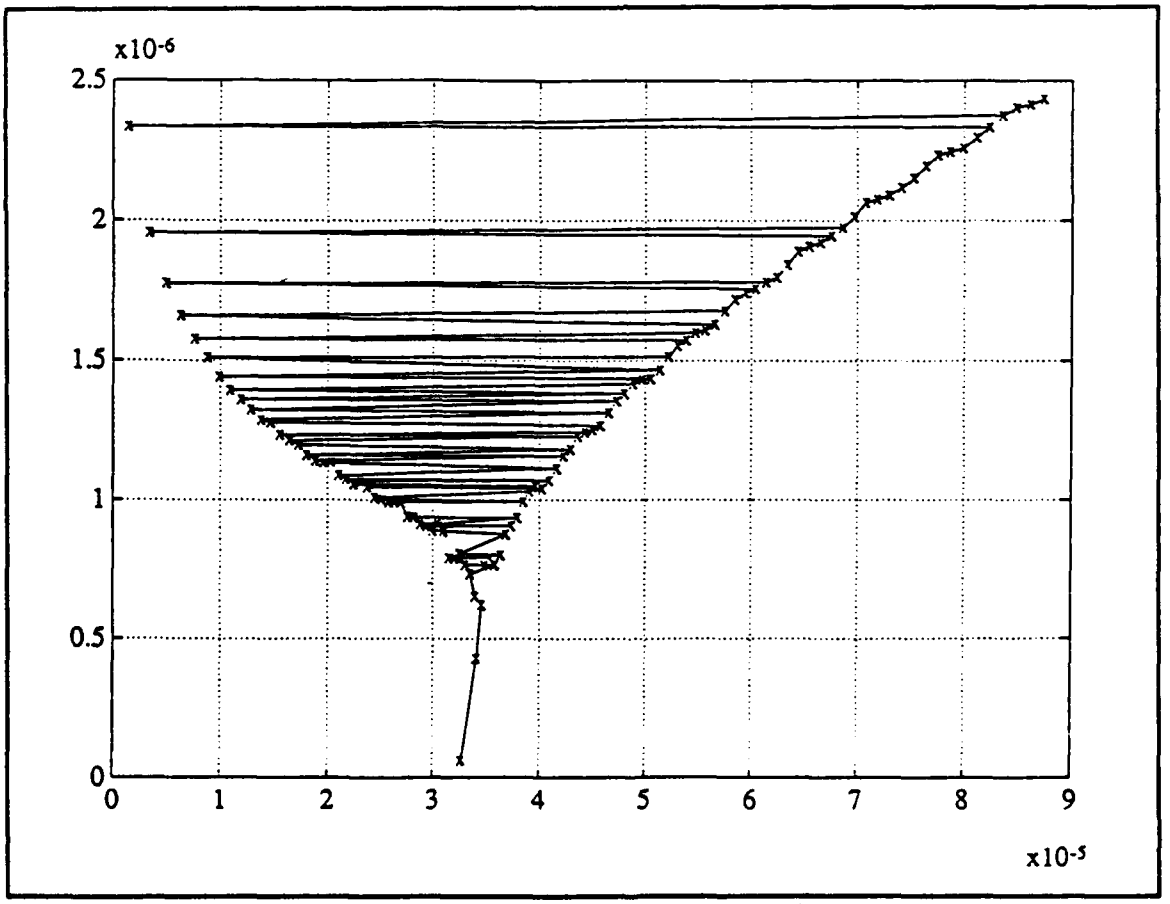


Figure 2. q-eigenvalues in the q_{11}/t_1 plane, connected in the order of ascending attenuation rate (20 m duct).

many pairs of consecutive modes in the upper part of this figure, it appears that the intuitive approach of starting with the mode of the lowest attenuation and continually looking for the mode with the next higher attenuation rate may not be the most efficient strategy. Furthermore, there seems to be a fork near the middle of the figure which suggests that there can be more than one constant phase line passing through all the zeros of the mode function. These problems suggest that, in order to design an efficient mode search strategy, the behavior of the mode function in the complex q_{11}/t_1 plane has to be investigated more thoroughly.

B. PHASE LINE TRACKING

The M-Layer program follows a line along which $\mathrm{Im}\{D(q)\} = 0$ until a zero is encountered at its intersection with another line along which $\mathrm{Re}\{D(q)\} = 0$. Since there is no way to determine the phase of the constant phase line linking two adjacent zeros *a priori*, following these two types of somewhat arbitrary but easy to compute phase lines which have constant phases of 0 or π, and $\pi/2$ or $3\pi/2$, respectively, are still the best way to program a computer to locate the zeros systematically. It is thus necessary to find out the actual distribution of these constant phase lines in the complex q_{11}/t_1 plane.

1. Downward Tracking

In the original design, along the real q axis, M-Layer divides the search region into "contour rectangles," each of which spans 160 meshes horizontally. From the observed locations of the modes, it is found desirable to track and plot the constant phase lines along $\mathrm{Im}\{D(q)\} = 0$ and $\mathrm{Re}\{D(q)\} = 0$ over a little more than six "contour

rectangles", roughly three to the right and three to the left of the starting real q coordinate given by Eq. (10), which is a variable called q_{test} in M-Layer. The subroutine FNDMOD and FZEROX are modified to track these constant phase lines. A listing of these modified routines for tracking the lines Im{D(q)} = 0 beginning from the top edge and moving downwards into the search region is included in Appendix B. Their flow charts are given in Appendix C. Only minor changes are required to track the type of lines Re{D(q)} = 0, or to track these lines from the bottom of the search region upwards. The program first searches for a sign change in Im{D(q)} or in Re{D(q)} along the top edge of the search region, starting at the coordinate -512 mesh sizes from q_{test}, until a distance of 1024 mesh sizes is covered. When a sign change is observed, a constant phase line is recognized as passing through the particular mesh square and the program writes the complex q value of the lower-left corner of this mesh square into an ASCII file identified by whether the mode function is real or imaginary along the line, and the order this line is found. The program then moves from the top edge downwards to follow this constant phase line until it exits the search region. The q values of the lower-left corner of the mesh squares along this phase line are also written into the same ASCII file to be plotted later using MATLAB/386. The constant phase lines for the mode functions of the 2, 4, 6, 8, 10, 20, 30 and 40 meter evaporation ducts are obtained and plotted. They are included in Appendix D. Tracking and plotting these constant phase lines are extremely time consuming. The CPU time required to track the desired constant phase lines for each duct using an Intel 80486 based PC running at a 33 MHz clock rate is listed in Table 1. The time required to plot those lines for each duct using an Intel 80386 based PC running

at a 16 MHz clock rate is also listed.

TABLE 1

CPU Time for Tracking and Plotting Constant Phase Lines

duct height (m)	tracking Im{D(q)} = 0 (hr:min:sec)	tracking Re{D(q)} = 0 (hr:min:sec)	plotting both lines (hr:min:sec)
2	1:47:39	1:46:23	0:21:28
4	2:47:41	2:48:50	0:23:17
6	3:43:54	3:45:11	0:25:42
8	4:25:49	4:23:41	0:29:34
10	6:08:14	6:11:03	0:32:25
20	8:26:58	8:24:22	0:34:21
30	9:08:14	9:06:45	0:36:49
40	8:39:04	8:37:54	0:36:53
Total	45:07:33	45:04:09	4:01:29

The constant phase line plot for the mode function of the 20 m evaporation duct is given in Fig. 3. The solid lines represent those having Im{D(q)} = 0; the dotted lines represent those having Re{D(q)} = 0. Every intersection of these two types of phase lines is the location of a q-eigenvalue scaled by t_1, thus indicating the presence of a mode. Note that every solid line intersects with a dotted line except for many of those running from the top through the bottom of the search region. None of the solid lines intersect with other solid lines, neither do the dotted lines. This is seen clearly in Fig. 4, which magnifies the lower center portion of Fig. 3. One feature that can be recognized

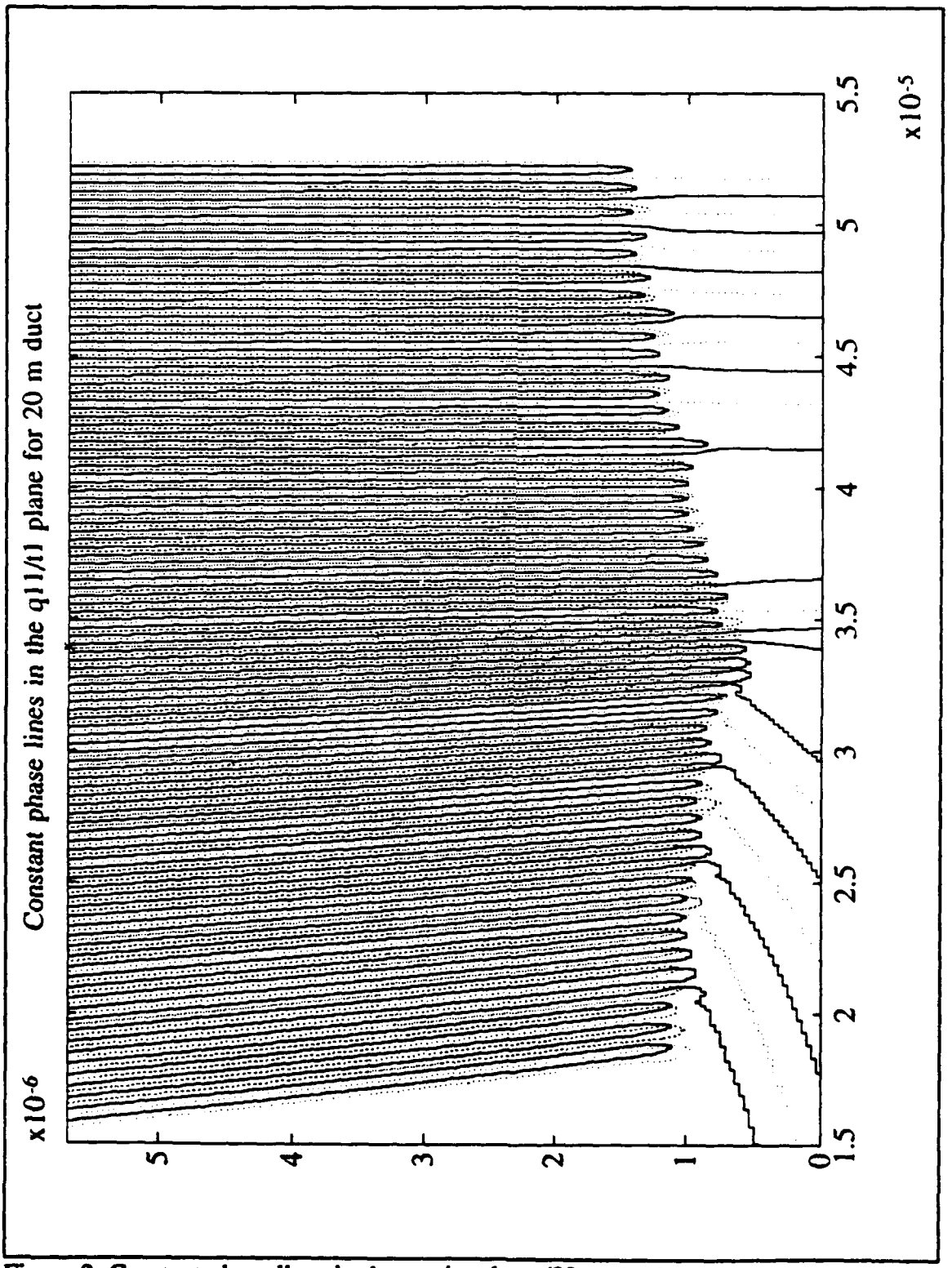


Figure 3. Constant phase lines in the q_{11}/t_1 plane (20 m duct).

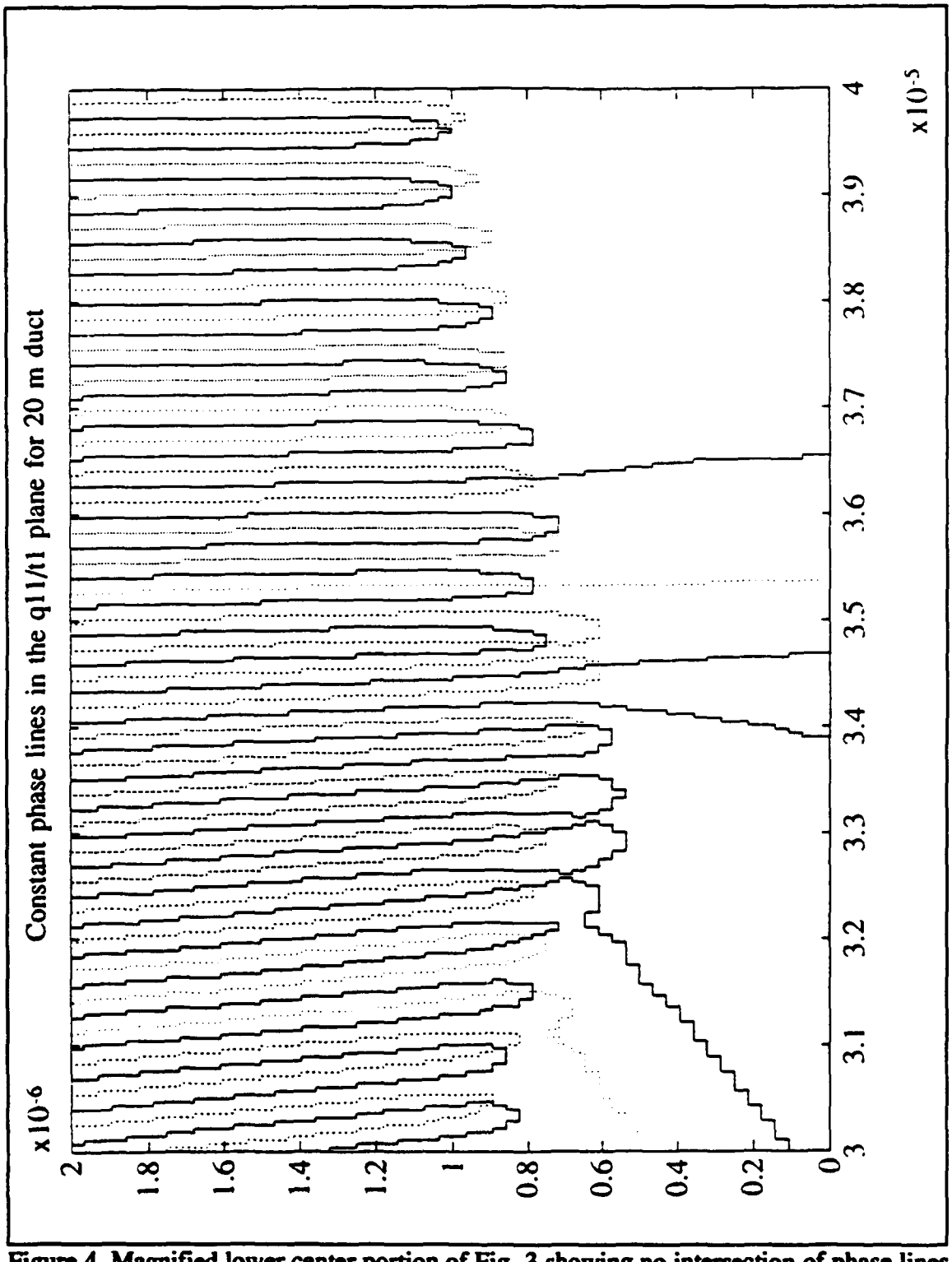


Figure 4. Magnified lower center portion of Fig. 3 showing no intersection of phase lines of the same type.

immediately in all the plots in Appendix D is that the starting real q coordinate given by Eq. (10), which is marked by an "x" along the top edge of each plot, correlates very closely with the mode of minimum attenuation.

2. Upward Tracking

One mode of low attenuation rate is missing from Fig. 2 compared to those found in Ref. 2. Several of them are also missing from those cases of greater duct heights plotted in Appendix D. It is evident that tracking the constant phase lines from the real q_{11}/t_1 axis upwards is necessary. For the 10 meter and the 20 meter ducts, Fig. 5 shows several constant phase lines starting out of the lower limit of the search region and returning to the lower-half complex q_{11}/t_1 plane. Fig. 6 shows the presence of such lines for the 30 meter and the 40 meter ducts. These constant phase lines have not been observed in those cases of lower duct heights.

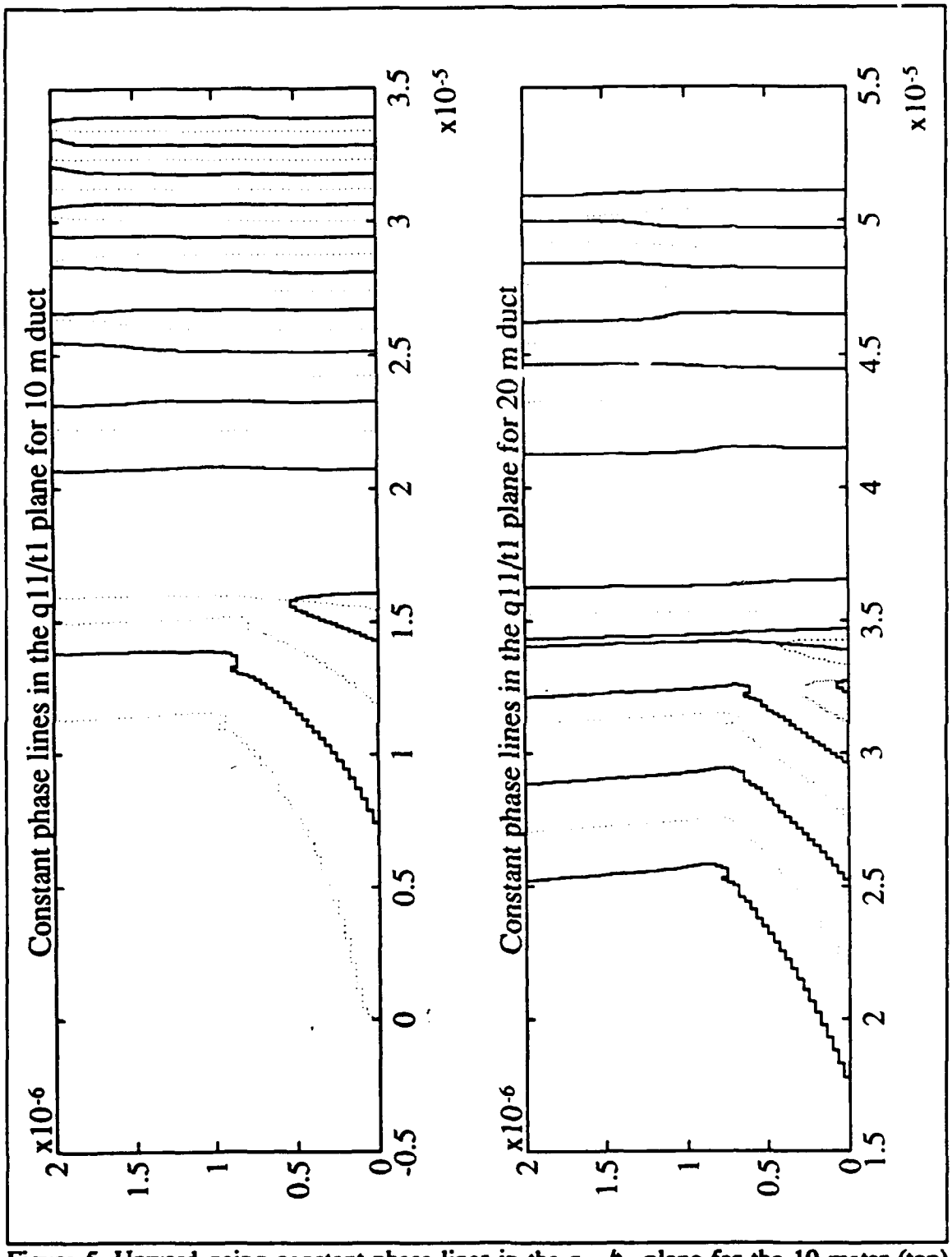


Figure 5. Upward going constant phase lines in the q_{11}/t_1 plane for the 10 meter (top) and 20 meter (bottom) ducts.

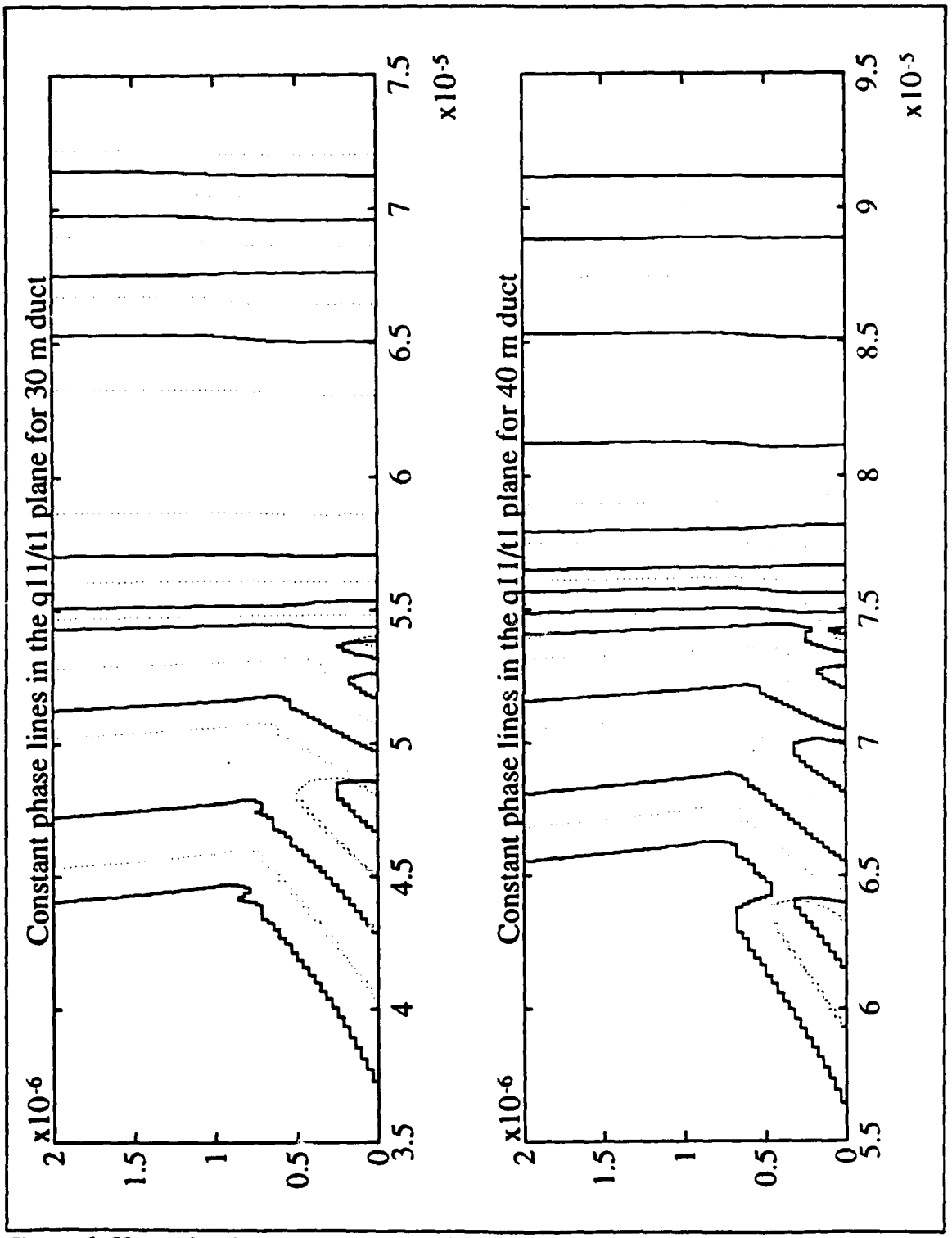


Figure 6. Upward going constant phase lines in the q_{11}/t_1 plane for the 30 meter (top) and 40 meter (bottom) ducts.

III. ANALYSIS AND CONCLUSIONS

A. MODE SEARCH PARAMETERS

The implementation of a mode search procedure requires several parameters. The search region has to be defined first. In M-Layer, the desired maximum range attenuation rate of the modes which support wave propagation is treated as an input parameter called a_{loss}, given conventionally in dB per km. This parameter determines the region over which the modes are to be searched as explained in Chapter I. In the complex q plane, under the assumption that both m_1 and β/k of interest are close to unity, the search region is bounded from above by the FORTRAN variable q_{top} given by

$$q_{top} = \frac{10^{-4} t_1}{k \times \log_{10} e} a_{loss} . \tag{12}$$

The location at which to start the mode search, q_{test}, is another parameter determined by M-Layer. Based on optical considerations, the M-deficiency, d_{mmin} of Fig. 1, which is the amount of decrease of the modified refractivity from its value on the sea surface to its minimum value in the air, is a measure of the capability of the duct to trap electro-magnetic waves. As explained also in Chapter I, this quantity determines the location for the program to start searching for modes. The validity of the optical argument and the usefulness of Eq. (10) have been proved by this work, as pointed out in Chapter

II.

To search for zeros of the mode function, M-Layer first covers the search region with identical mesh squares. It then follows the lines of the type $\mathrm{Im}\{D(q)\} = 0$ through each mesh square, checking for indications that a curve $\mathrm{Re}\{D(q)\} = 0$ is entering the same mesh so that an intersection is possible. The term "mesh size" will denote the size of the edges of the mesh squares. It is the size of the step taken by the program to advance through the search region. It also determines the initial resolution of the location of a mode. The choice of the mesh size should strike a balance between the desire for a speedy completion of the search process and the requirement in mode locating accuracy. As reported in Ref. 2, for all the evaporation ducts considered, the choice of a mesh size equivalent to an attenuation rate of 1/32 dB per km appears to be optimal, that is, all modes can be found for all cases investigated in Ref. 2 without experiencing extraordinarily long computation time. Consider this mesh size as the default and call it q_d in the complex q plane. Under the same assumption for deducing Eq. (12), the value of q_d/t_1 at 9.6 GHz equals 3.58×10^{-8}. Figure 7 shows the separation between two adjacent constant phase lines of the type $\mathrm{Im}\{D(q)\} = 0$, indicated in the figure as solid lines, along the top edge of the search region for the 20 meter duct. The minimum of these separations in terms of q_{11}/t_1 is about 2.8×10^{-7}, which corresponds to a spacing of a little less than eight mesh squares apart. The minimum separation between adjacent $\mathrm{Im}\{D(q)\} = 0$ and $\mathrm{Re}\{D(q)\} = 0$ lines is thus about four meshes. Constant phase line separations for all evaporation ducts considered are included in Appendix E. Note that the minimum separation is almost constant for ducts higher than 20 meters. It increases

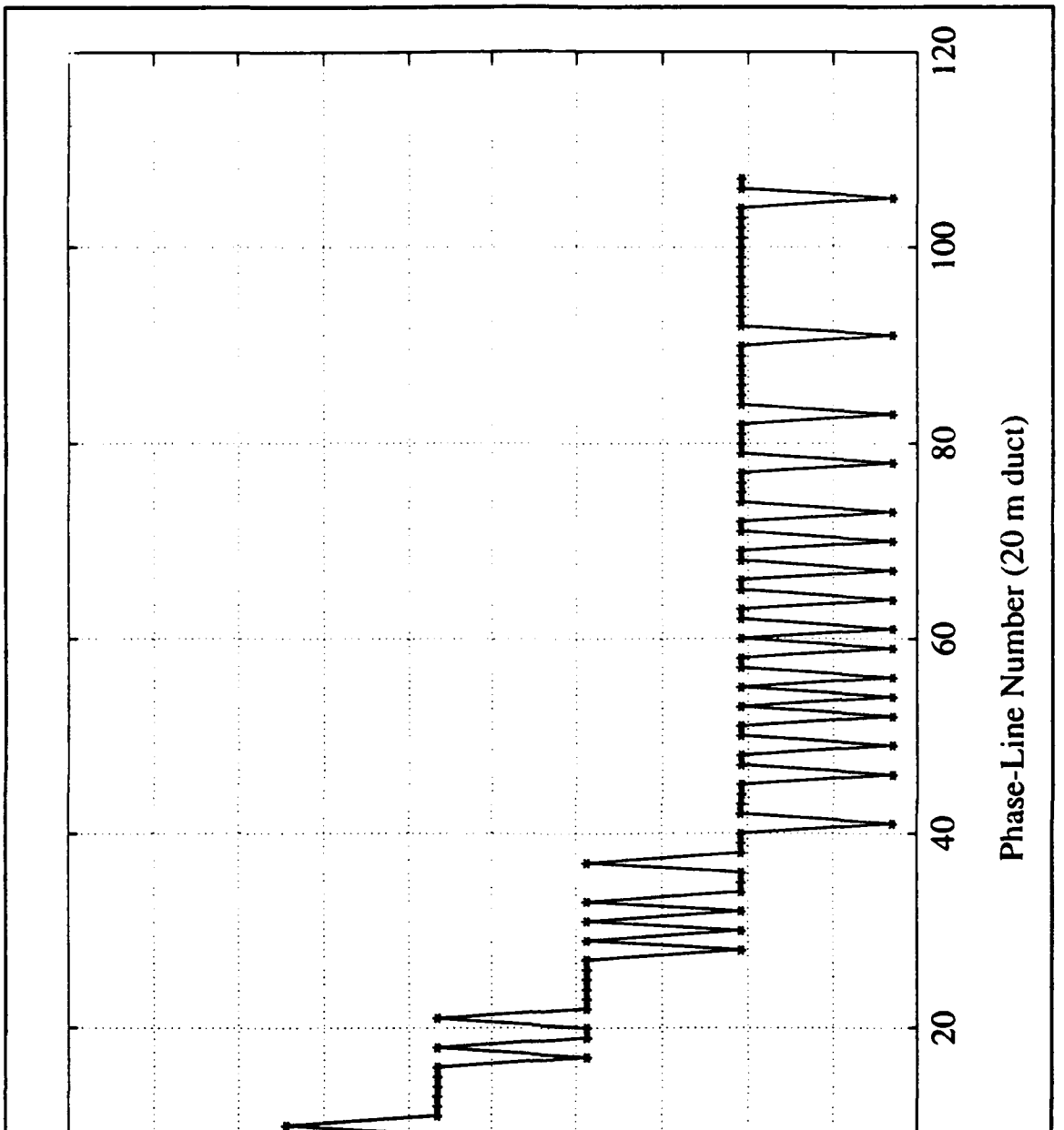


Figure 7. Separation of Im{D(q)} = 0 lines along the top edge of the search region (20 m duct).

slightly as duct height is decreased.

As the program follows an $\mathrm{Im}\{D(q)\} = 0$ line into the search region, a neighboring $\mathrm{Re}\{D(q)\} = 0$ line moves closer and enters into the same mesh square. This causes the program to invoke the root finding routine to determine the possible locations of zeros of the mode function within this mesh square. If the mesh size is too large and the $\mathrm{Re}\{D(q)\} = 0$ line enters the mesh square long before the actual intersection takes place, the root finding routine will be prematurely invoked many times and the probability of producing false modes is increased. This explains why a larger mesh size sometimes will increase the execution time.

Given these three parameters q_{top}, q_{test}, and q_d, a mode search strategy which does without the "contour rectangles" is proposed. It should improve the efficiency of M-Layer.

B. MODE SEARCH STRATEGY

From the constant phase line plots of Appendix D, a strategy to search for the mode can be drawn: move along the top edge starting at q_{test}. Search first toward either the left or the right, then reverse course to search along the other direction. After the search along the top edge is completed, search the lower edge from one end to the other. In what follows, the implementation of this strategy will be discussed.

1. Track Termination

The tracking of an $\mathrm{Im}\{D(q)\} = 0$ line naturally ends if the top edge or the bottom edge of the search region is reached. On the other hand, the search region is unbounded to the right and left of q_{test}. It is convenient to retain the feature in the

original program to set a limit on the number of steps allowed to follow a constant phase line. From Fig. 3, this limit can be set to 2.5 times the number of mesh squares between the vertical limits of the search region. This number equals $2.5 \times 32 \times a_{loss}$.

2. **Search Termination**

M-Layer has to determine that no more modes within the search region is to be found and to terminate the search for modes. When searching along the top edge of the search region for the constant phase lines $\mathrm{Im}\{D(q)\} = 0$, the separation between the real parts of the mode eigenvalues is of interest. Figure 8 shows the separation in the real part of neighboring q-eigenvalues scaled by t_1, plotted against their locations q_m/t_1 along the real q_{11}/t_1 axis for the case of the 20 m duct. Similar plots for all duct heights are grouped in Appendix F. Excluding the lowest point which involves the mode obtained via searching along the lower edge, the separation between neighboring modes is erratic with an upward trend away from the center of the figure, which is close to the search starting position.

The increase in distance between two modes towards the end of the range searched makes it difficult to implement an adaptive mechanism to terminate the search. On the other hand, Fig. 3 shows that almost all phase lines entering the search region from the top that contain a mode within this region are bunched together. Therefore, the parameter n_{stop}, set equal to four and may be adjusted, can be used to stop the search along the top edge when four consecutive constant phase lines tracked turn up no mode.

The search along the real q axis can be confined to within the end points of the search along the top edge.

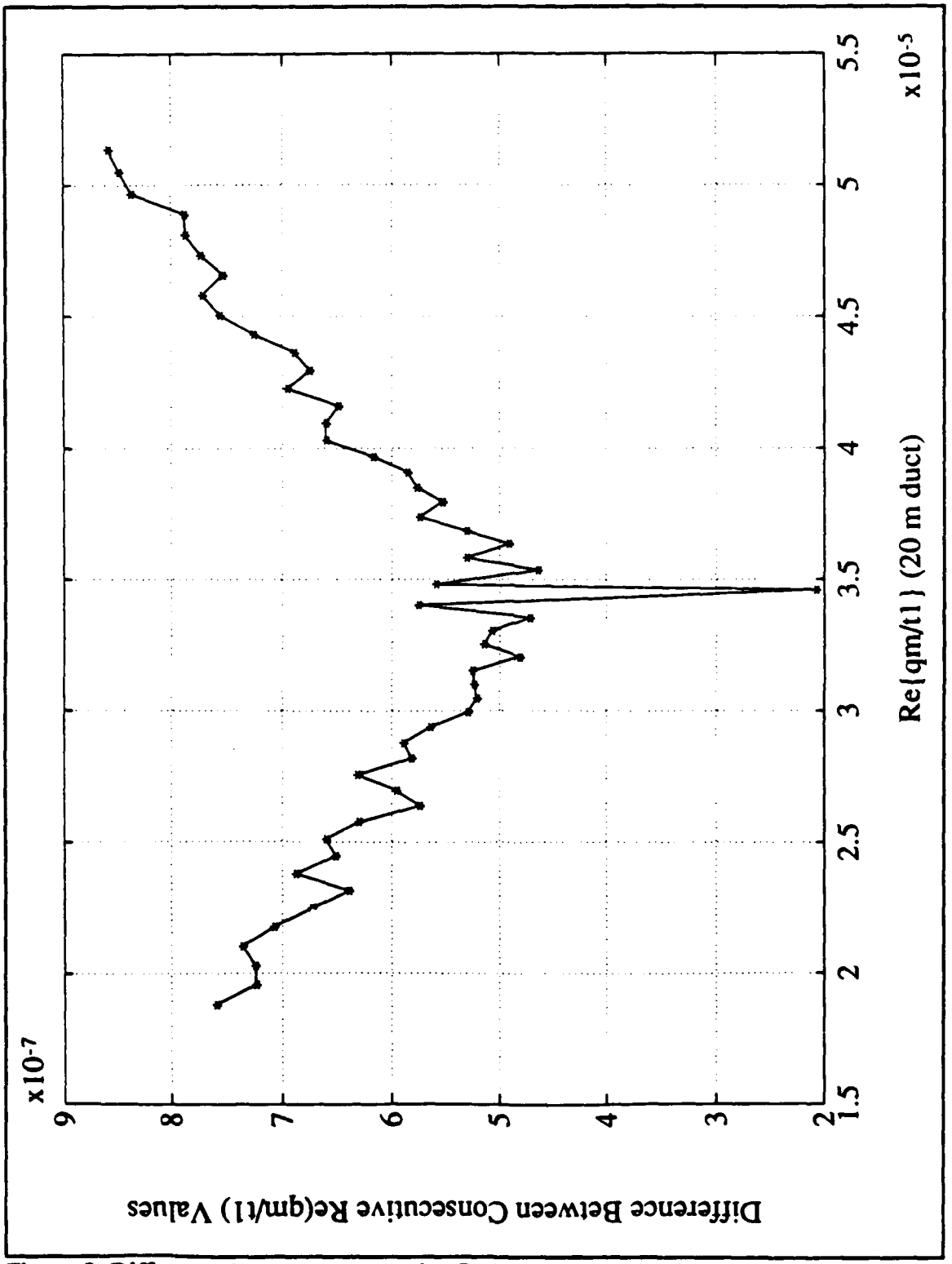


Figure 8. Difference between consecutive $Re(q_m/t_1)$ values (20 m duct).

3. Track Duplication Avoidance

When the program searches step by step along the top or bottom edges for sign changes in Im{D(q)} to start tracking the constant phase line, the exit of a constant phase line from a previous track will be encountered. Entering into the search region at such a location will simply retrace a constant phase line which has already been examined for the existence of a mode. Thus, the exit locations of the constant phase lines must be recorded and checked whenever a sign change in Im{D(q)} is found before deciding to follow the constant phase line into the search region. Along the top edge, a record of four updated most recent exit locations from the top edge should be adequate. The locations of exits from the bottom should be kept for use during the search along the lower edge of the search region. For this record, a dimension of 256 should be adequate for the current cases. But this dimension should be adjusted if other types of ducts are studied. A record of four of the most recent exit locations out of the lower edge should also be kept and checked to avoid duplicate tracking of the constant phase lines.

APPENDIX A: MODE LOCATIONS IN THE COMPLEX q_{11}/t_1 PLANE

This appendix contains figures of mode locations of evaporation ducts of 2, 4, 6, 8, 10, 20, 30, and 40 m heights plotted in the complex q_{11}/t_1 plane for easy comparison.

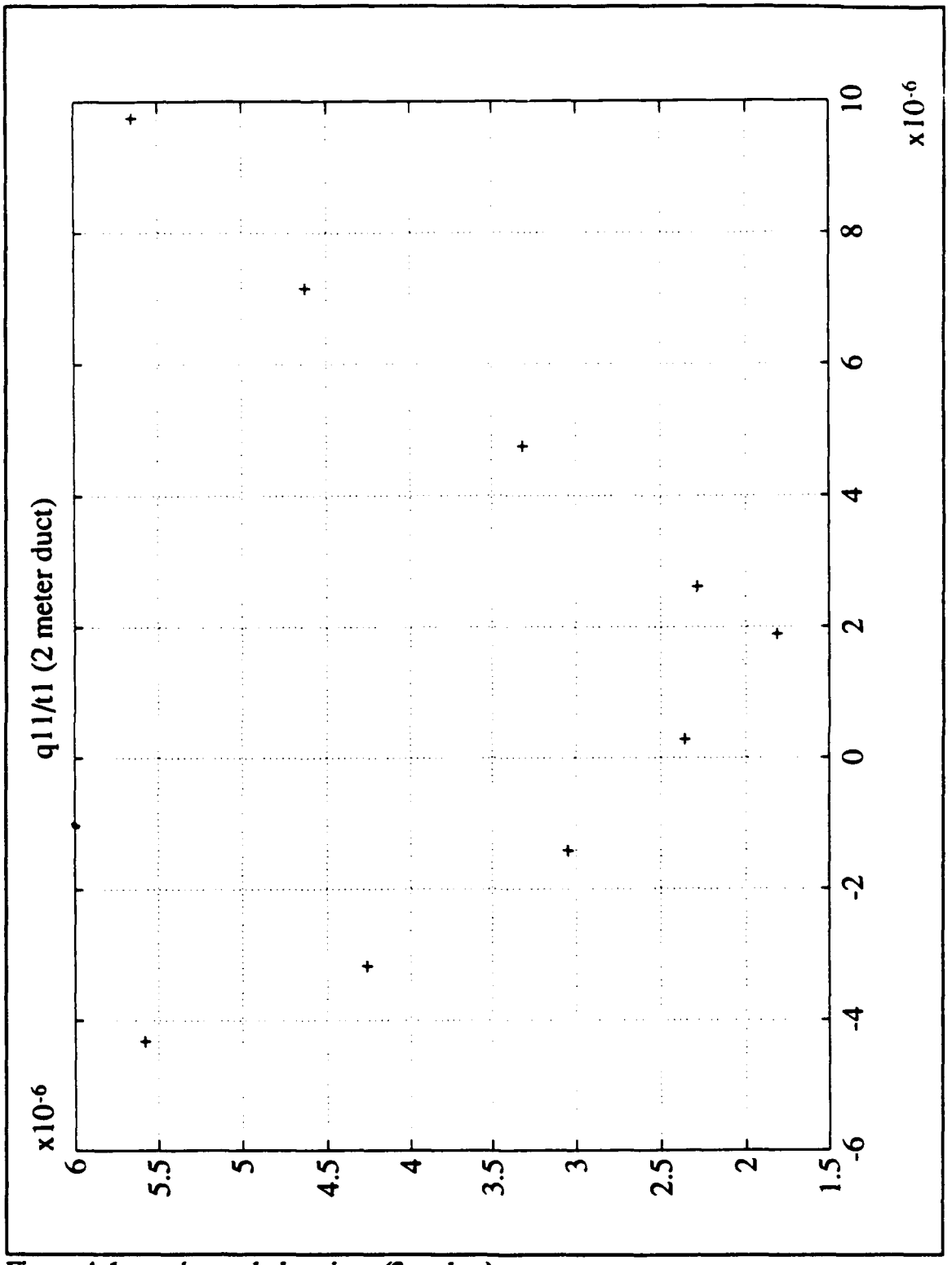


Figure A.1 q_{11}/t_1 mode locations (2 m duct).

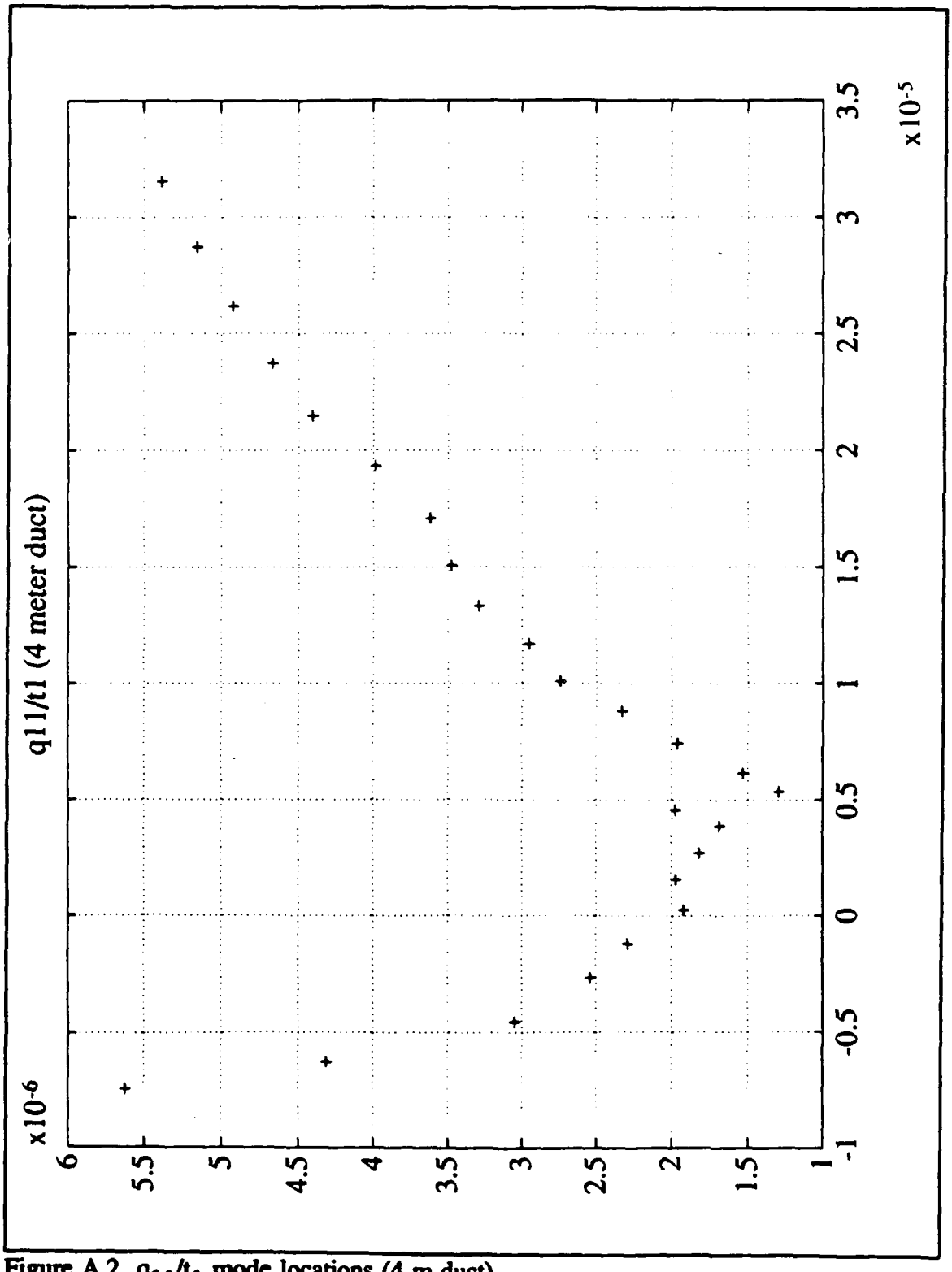


Figure A.2 q_{11}/t_1 mode locations (4 m duct).

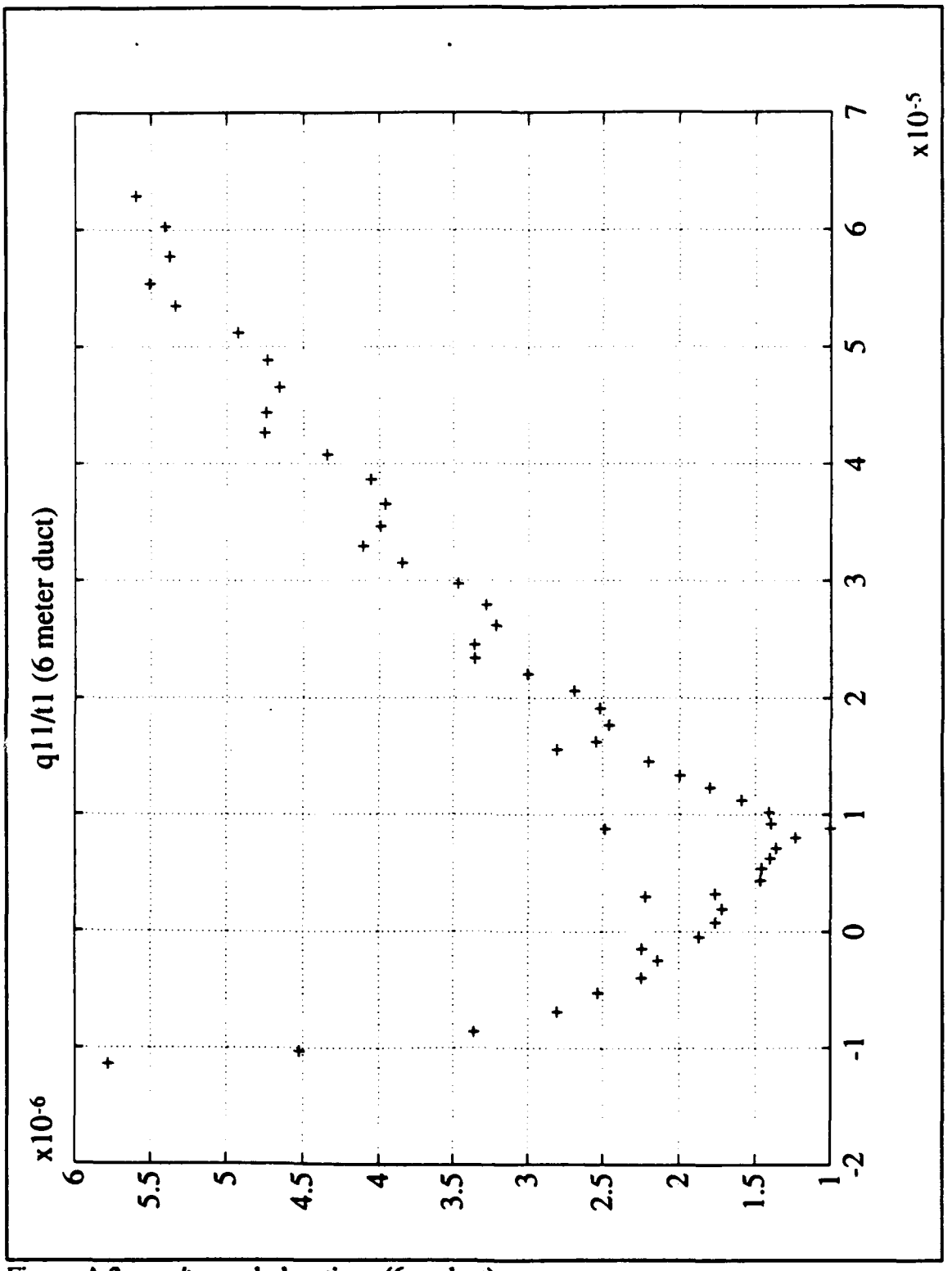


Figure A.3 q_{11}/t_1 mode locations (6 m duct).

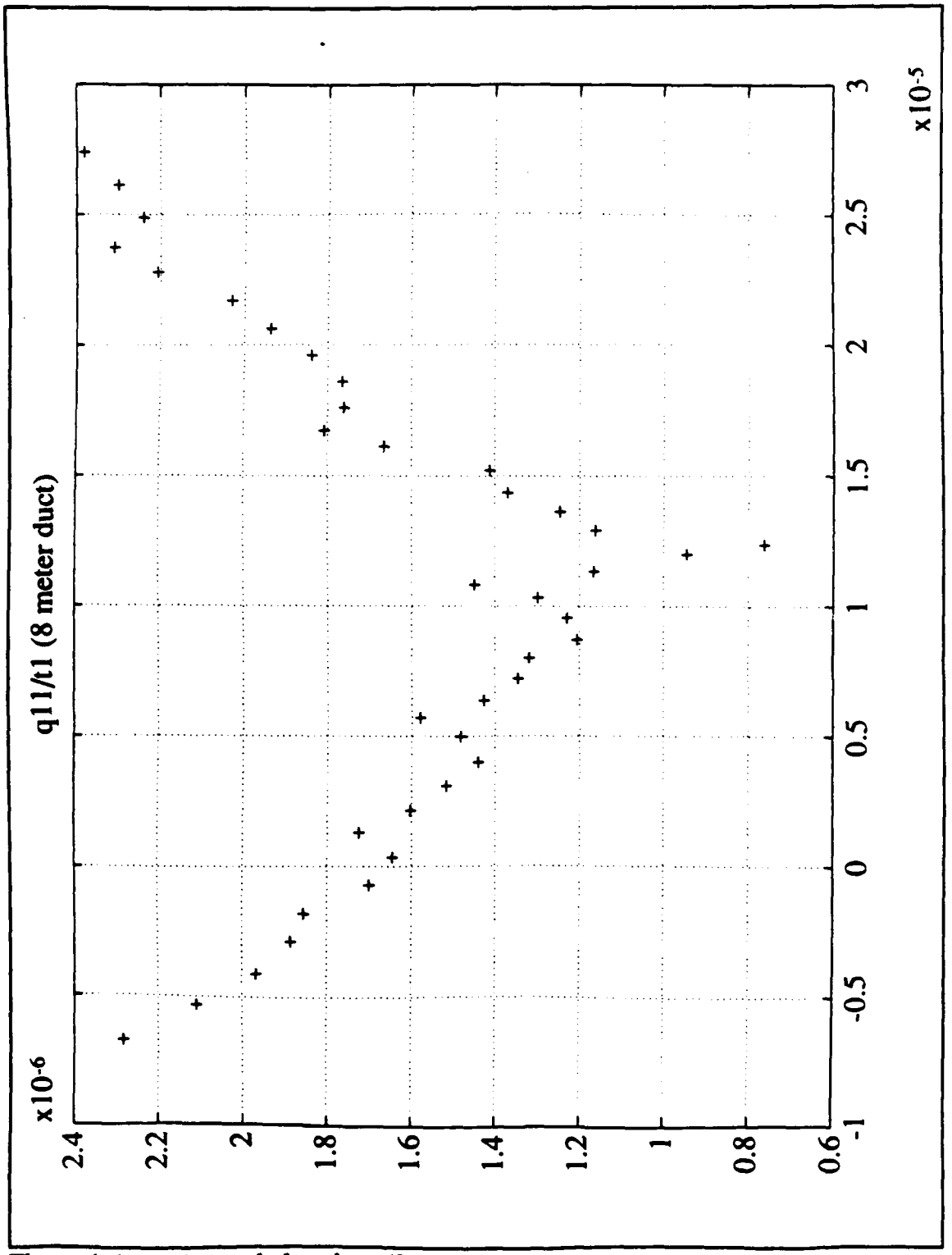


Figure A.4 q_{11}/t_1 mode locations (8 m duct).

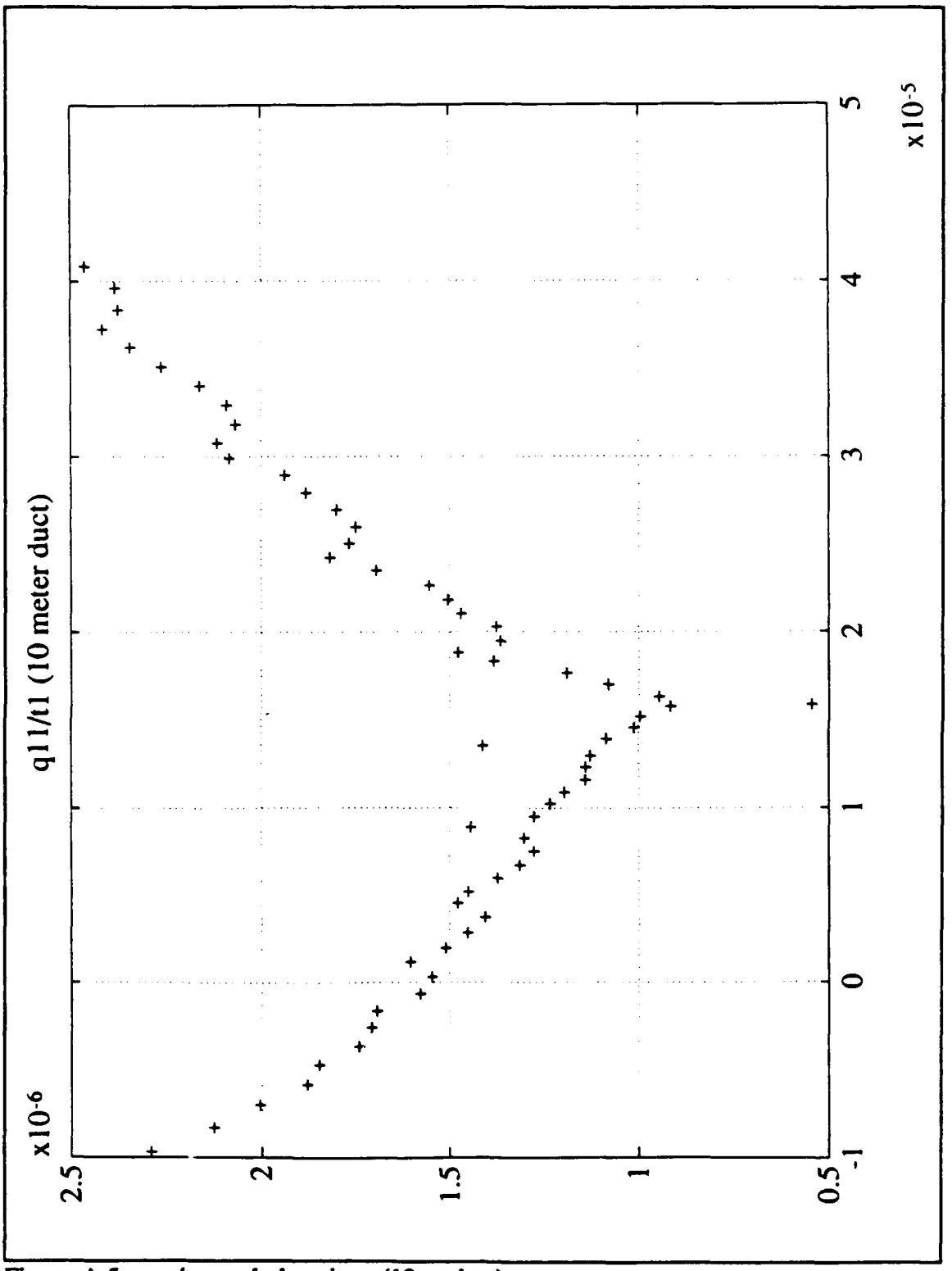


Figure A.5 q_{11}/t_1 mode locations (10 m duct).

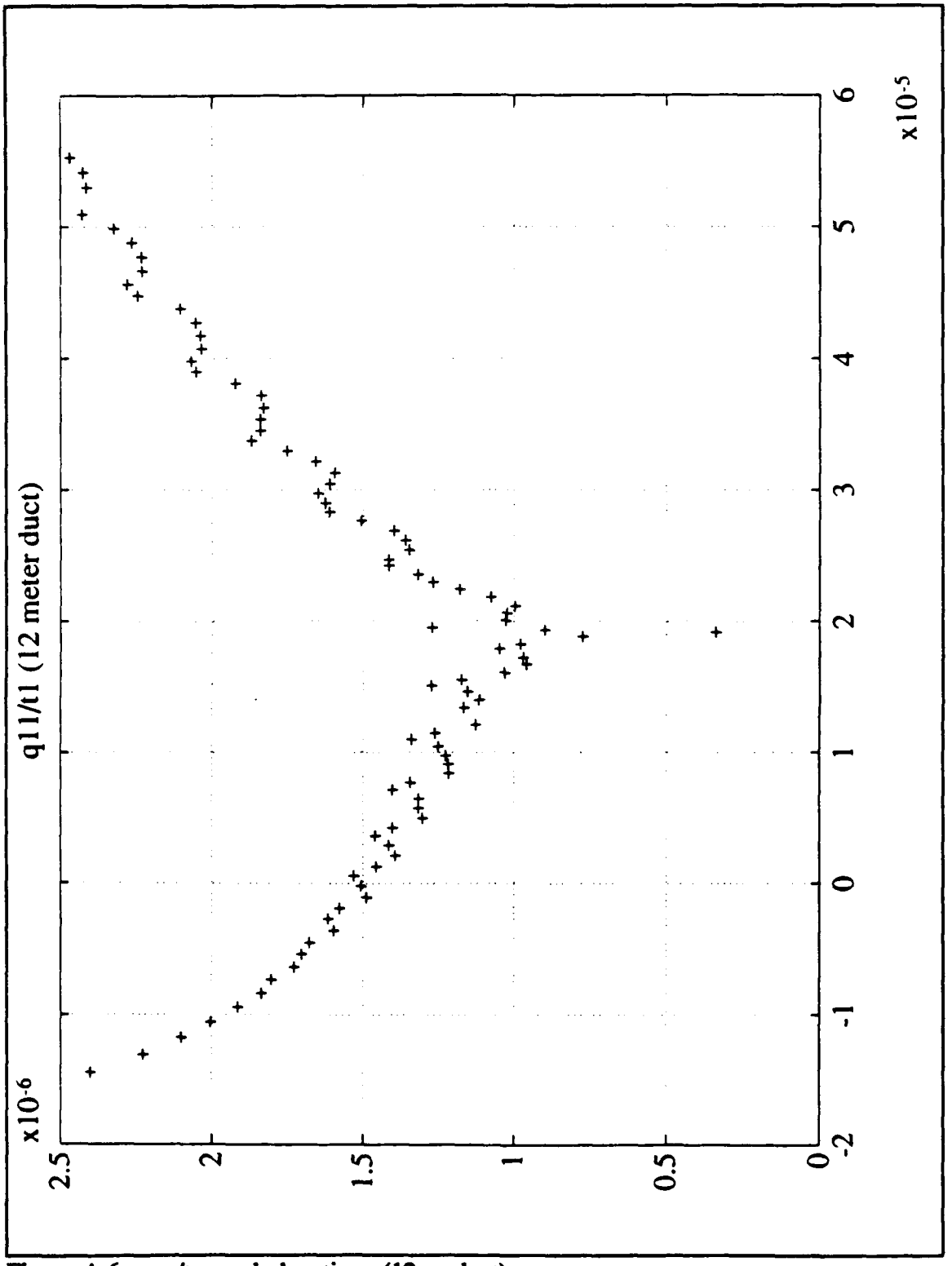


Figure A.6 q_{11}/t_1 mode locations (12 m duct).

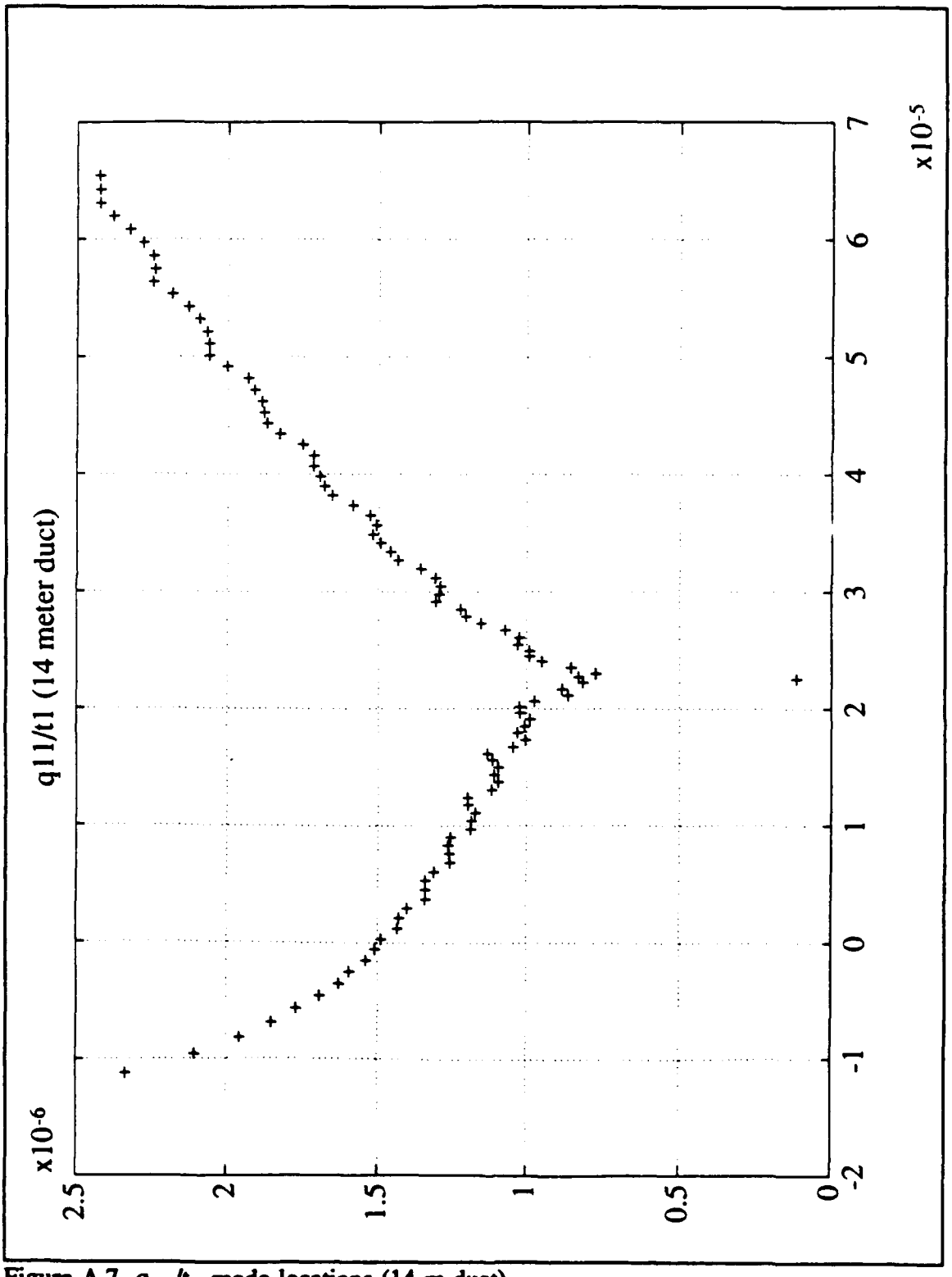


Figure A.7 q_{11}/t_1 mode locations (14 m duct).

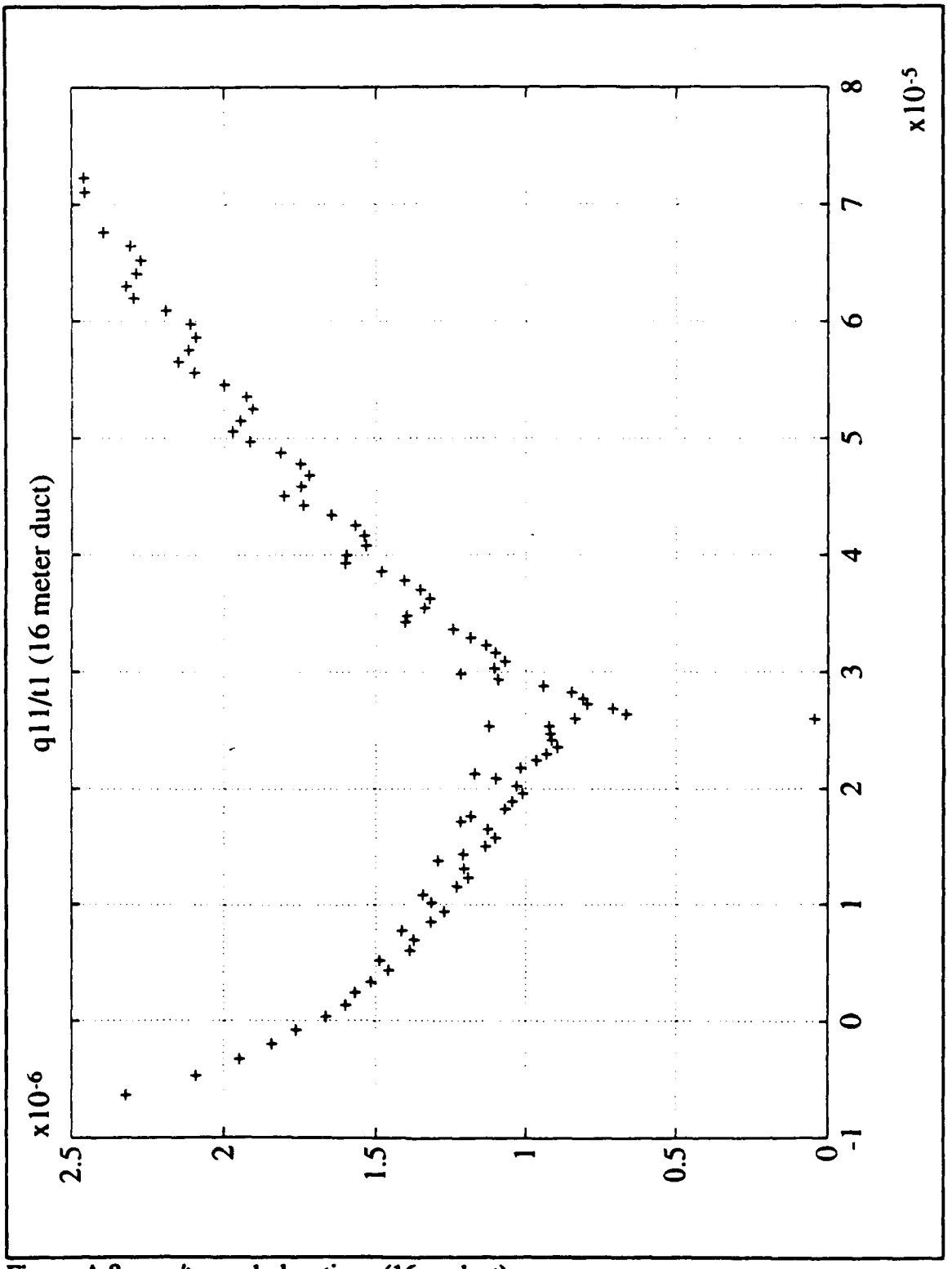


Figure A.8 q_{11}/t_1 mode locations (16 m duct).

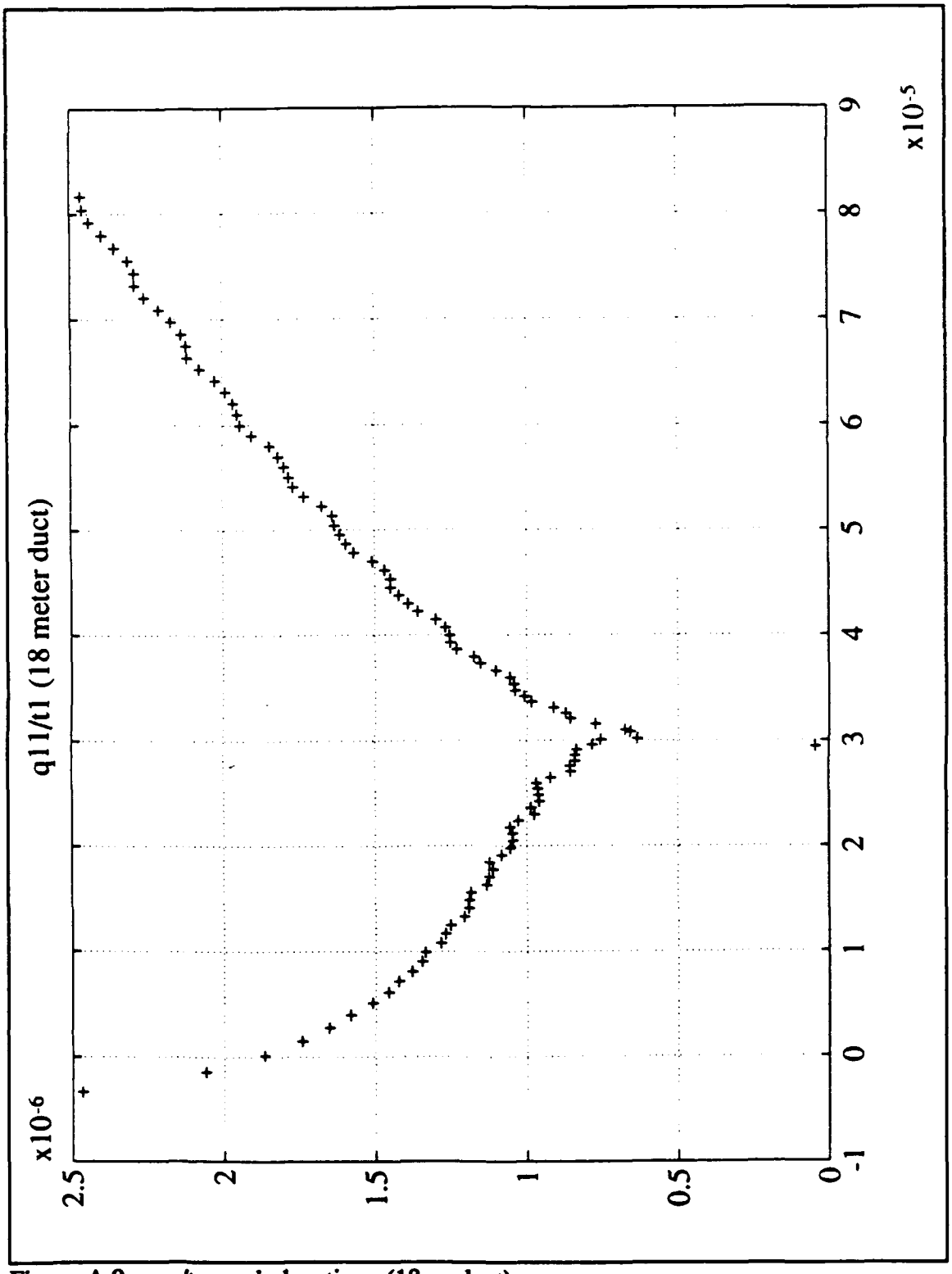


Figure A.9 q_{11}/t_1 mode locations (18 m duct).

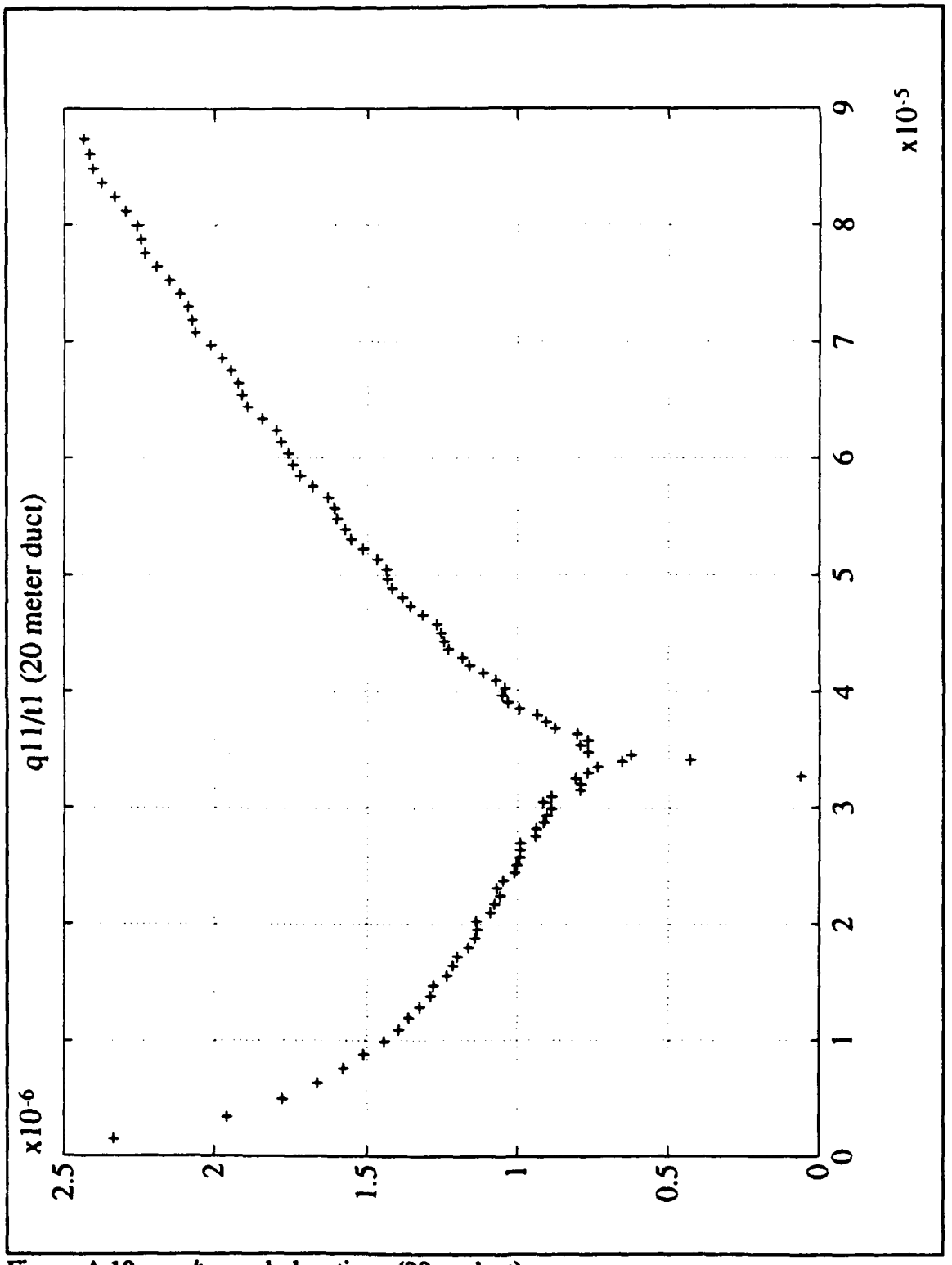


Figure A.10 q_{11}/t_1 mode locations (20 m duct).

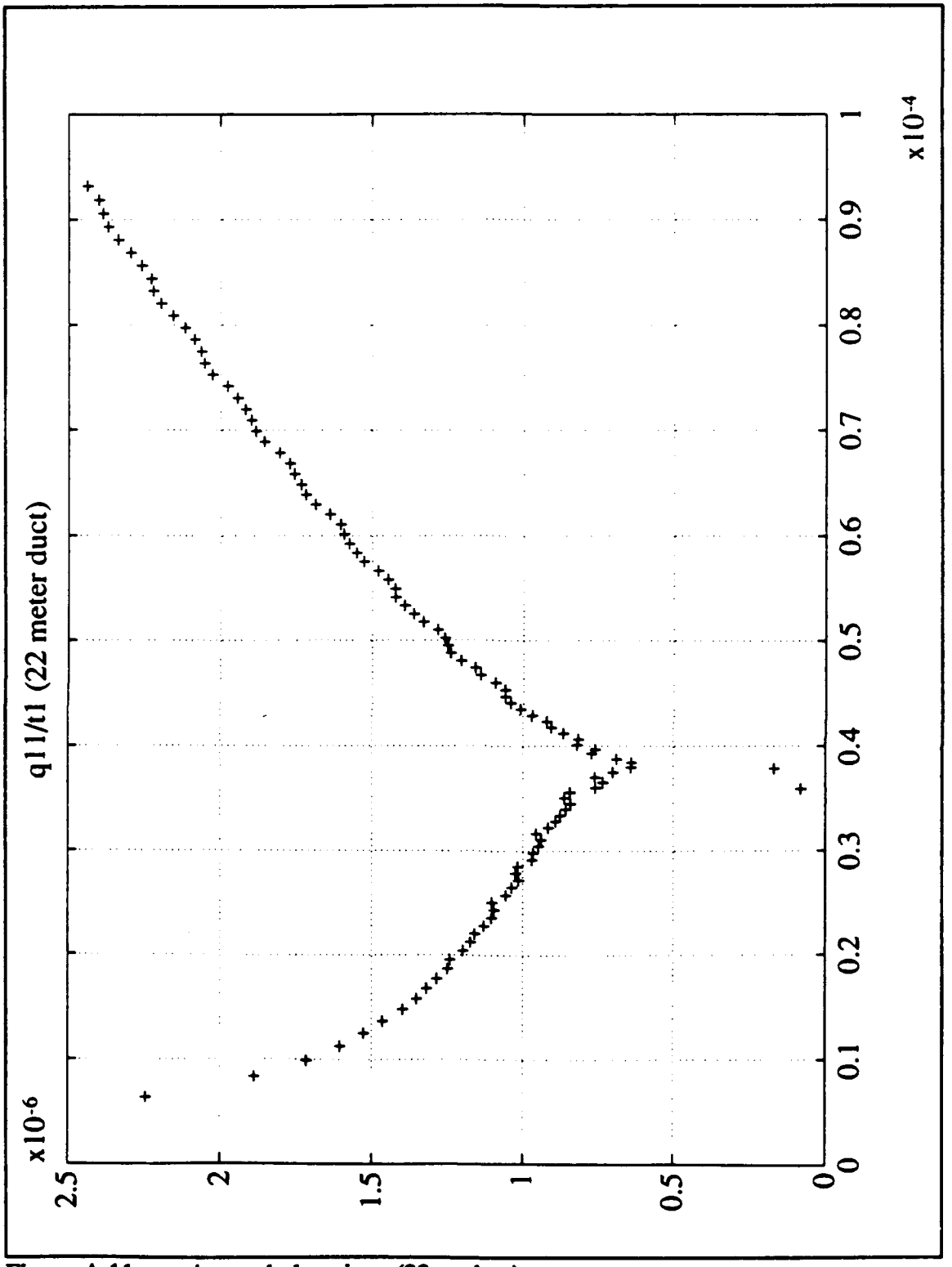


Figure A.11 q_{11}/t_1 mode locations (22 m duct).

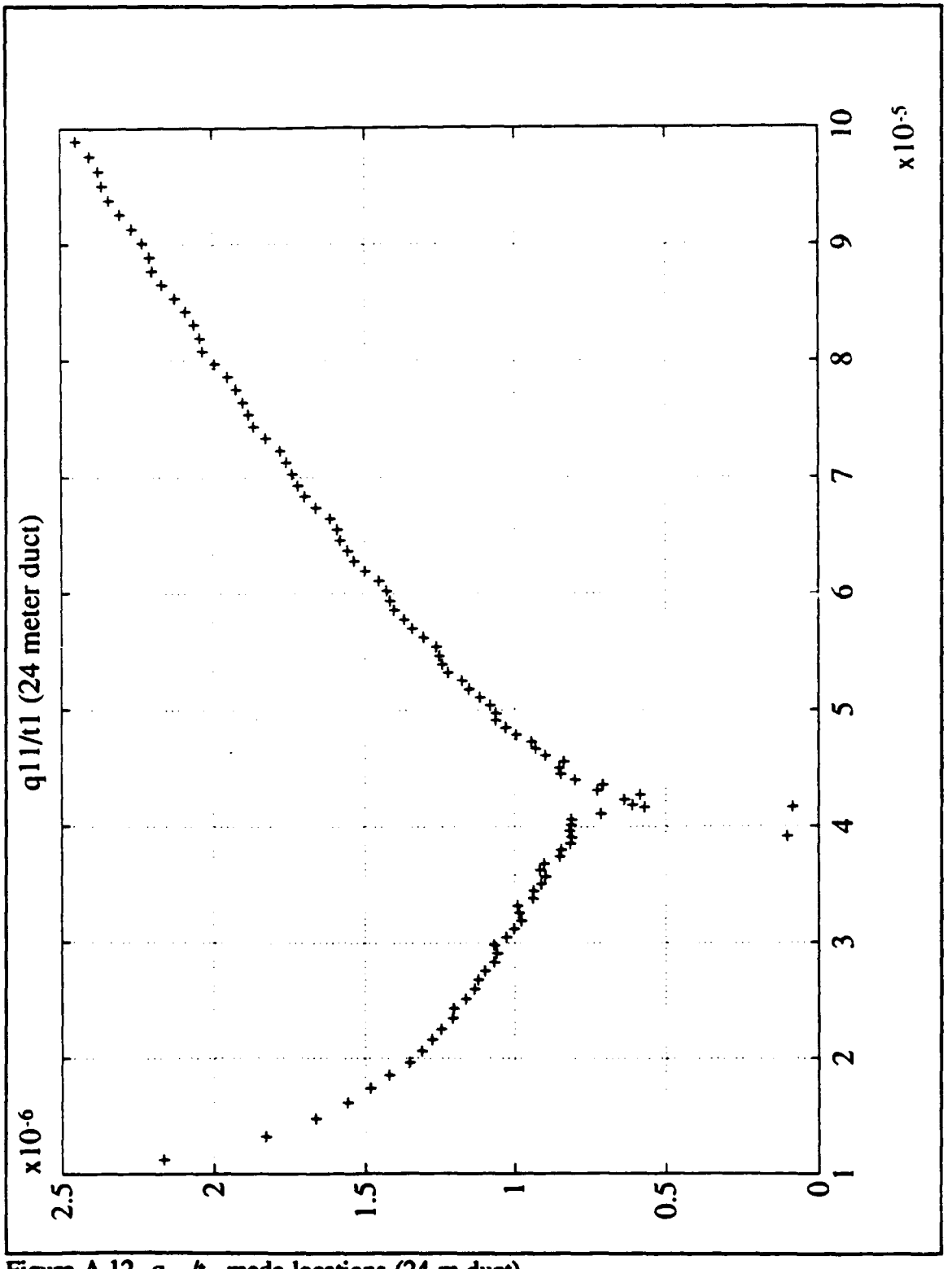


Figure A.12 q_{11}/t_1 mode locations (24 m duct).

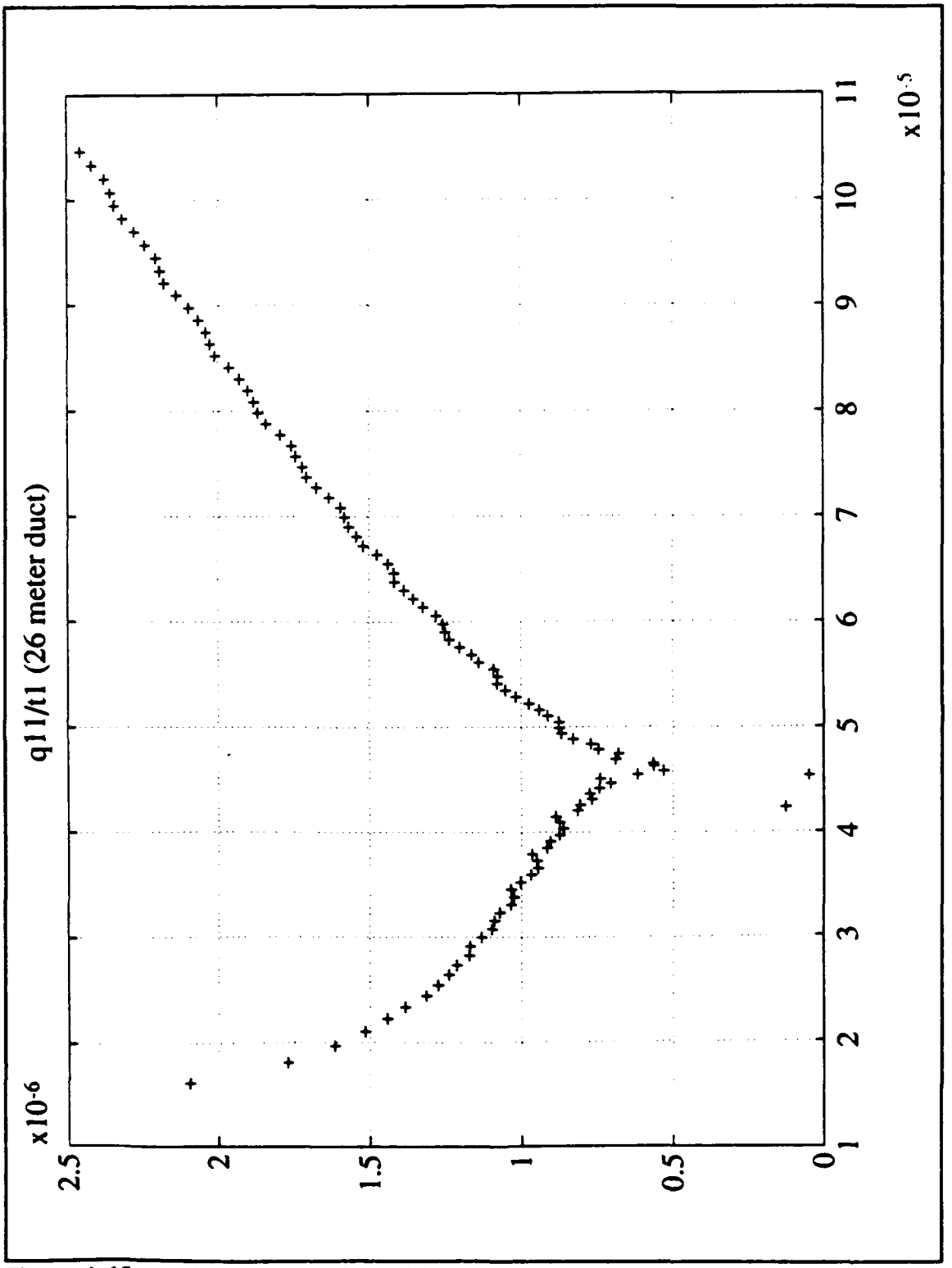


Figure A.13 q_{11}/t_1 mode locations (26 m duct).

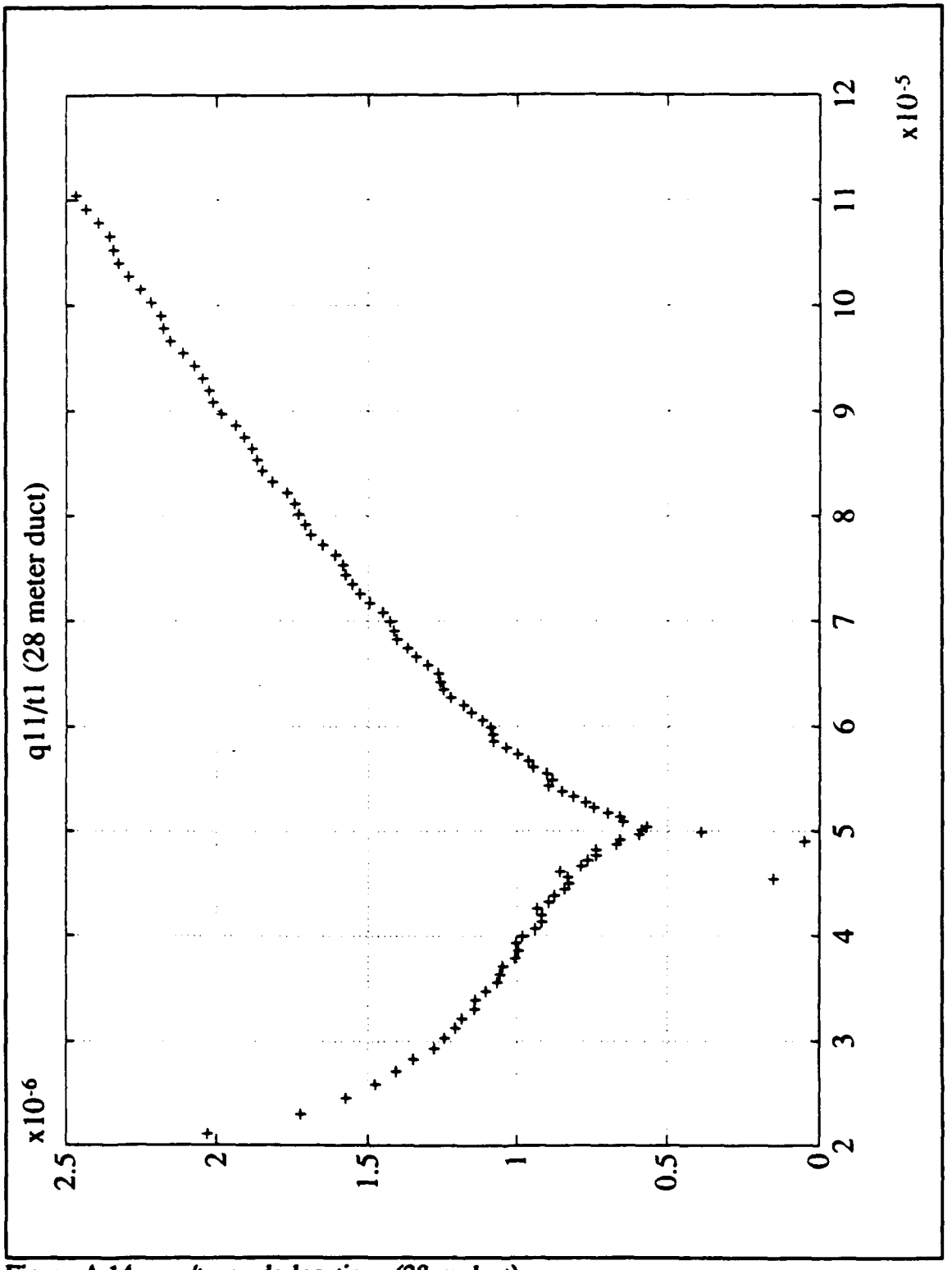


Figure A.14 q_{11}/t_1 mode locations (28 m duct).

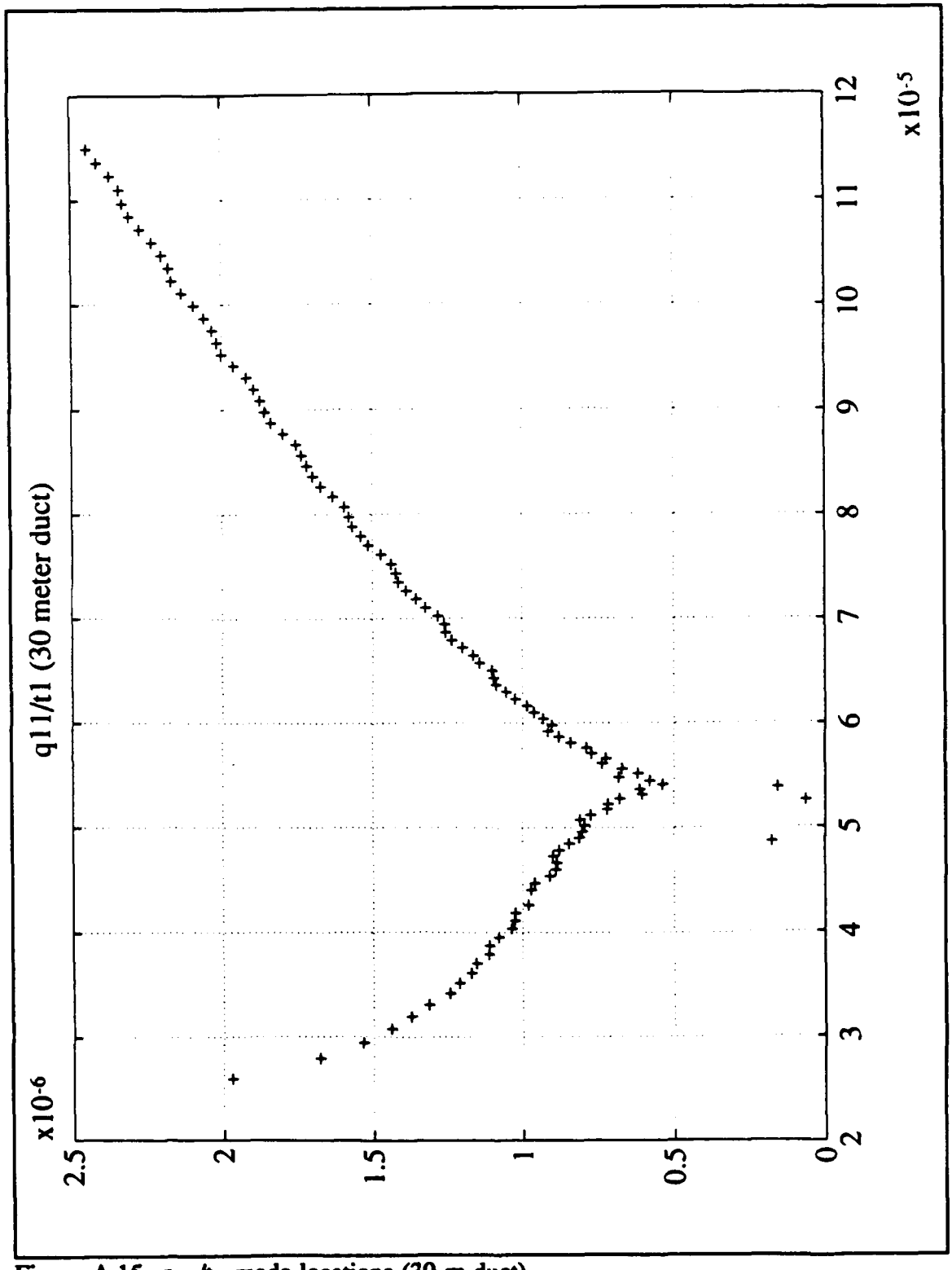


Figure A.15 q_{11}/t_1 mode locations (30 m duct).

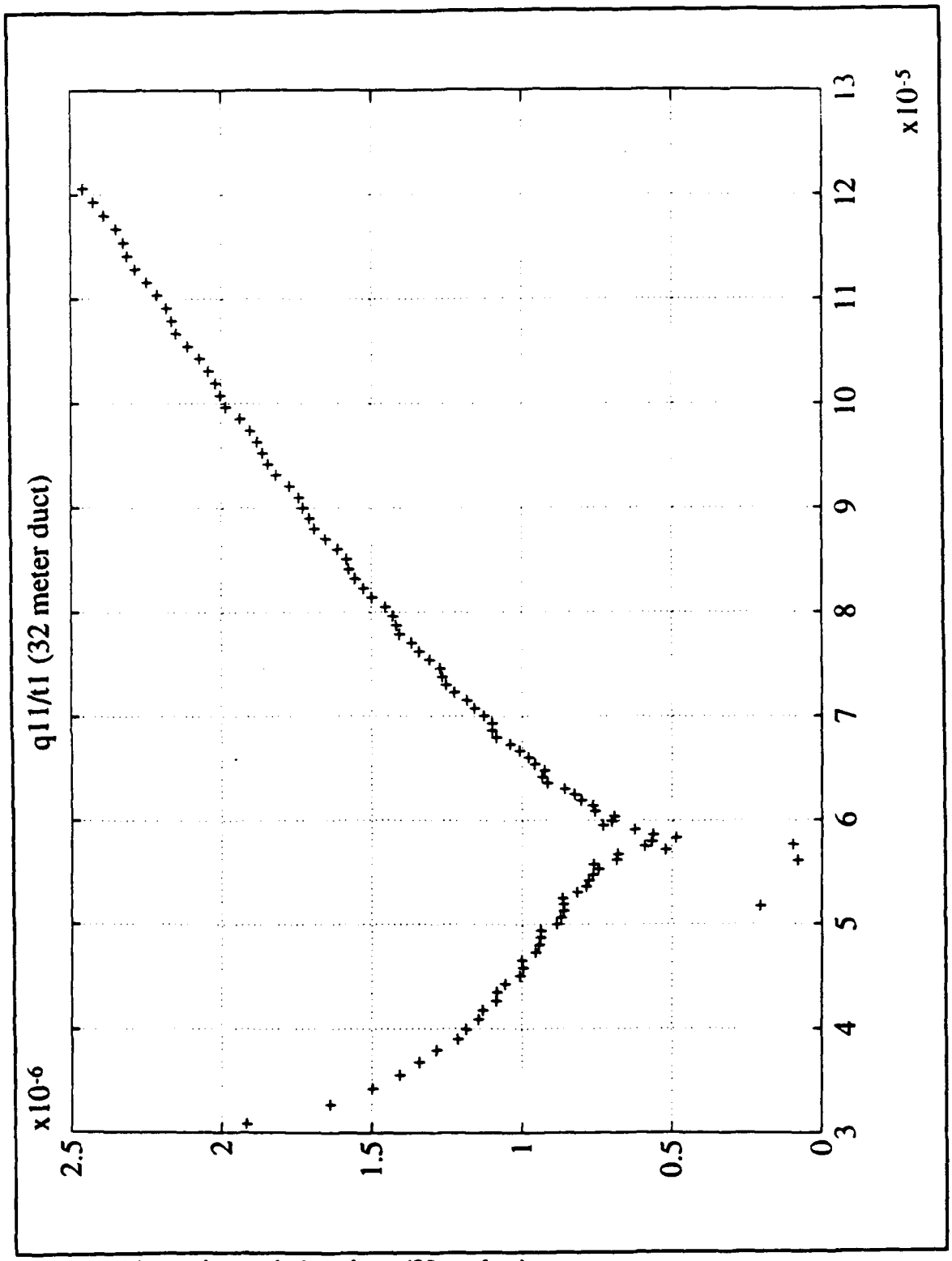


Figure A.16 q_{11}/t_1 mode locations (32 m duct).

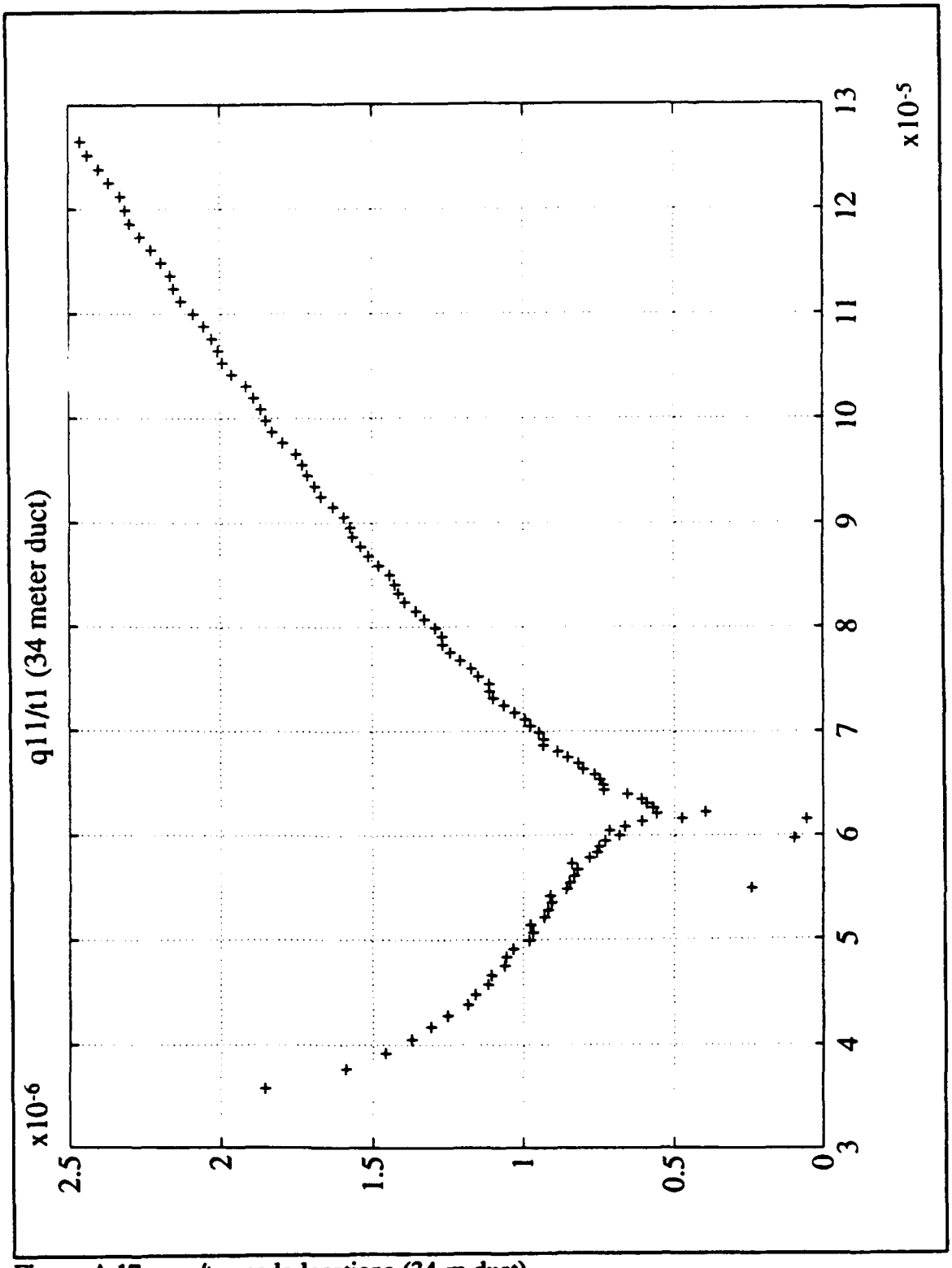


Figure A.17 q_{11}/t_1 mode locations (34 m duct).

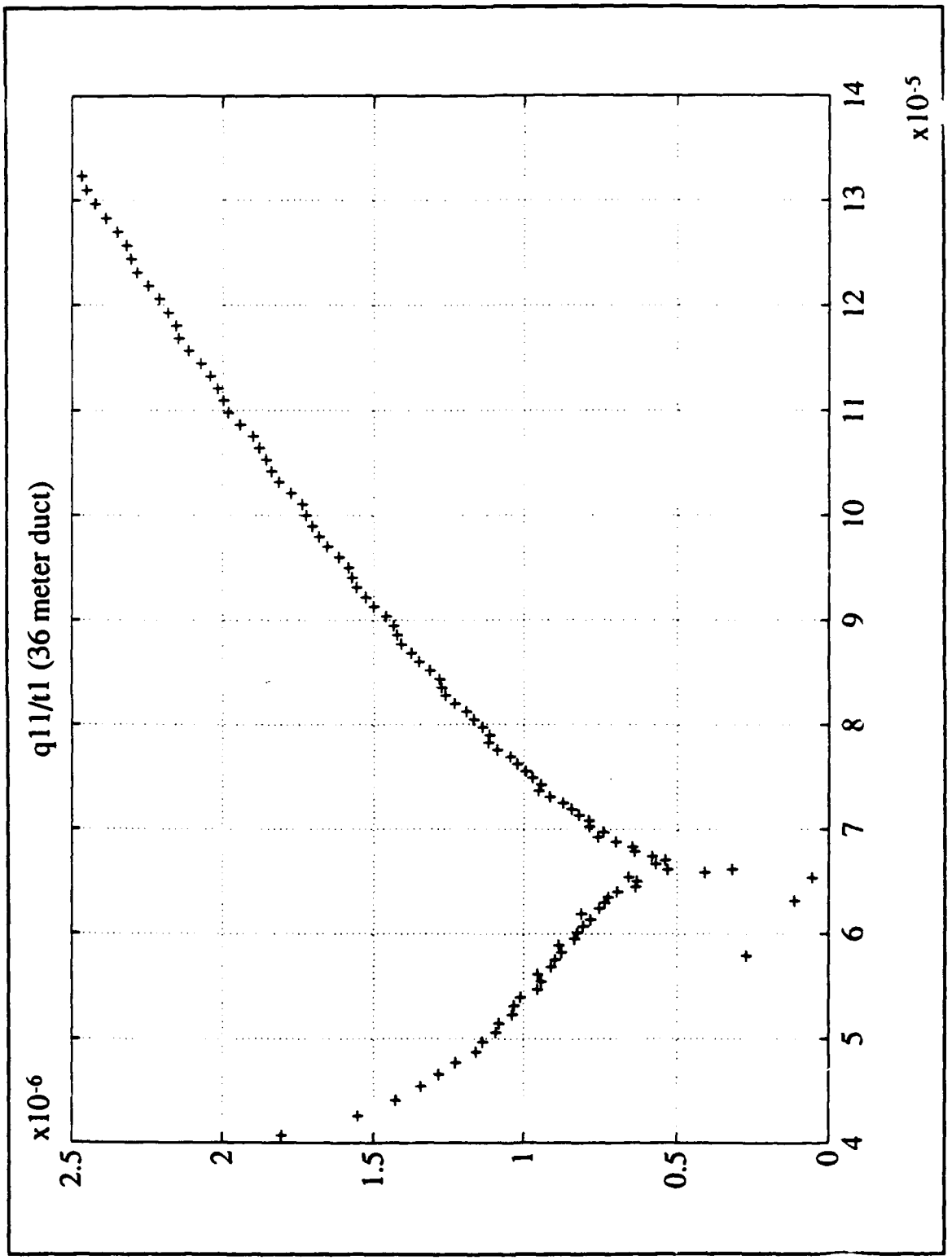


Figure A.18 q_{11}/t_1 mode locations (36 m duct).

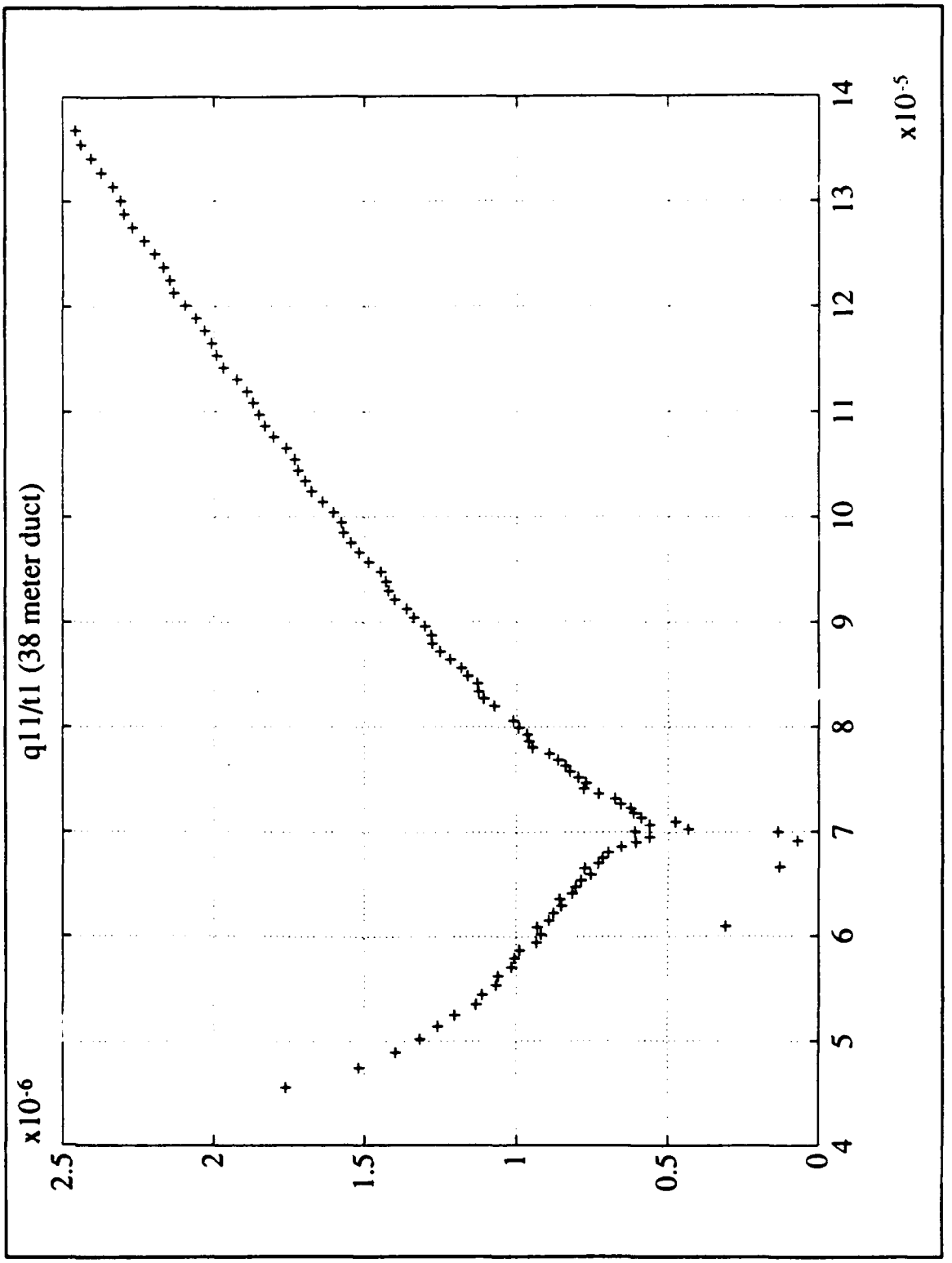


Figure A.19 q_{11}/t_1 mode locations (38 m duct).

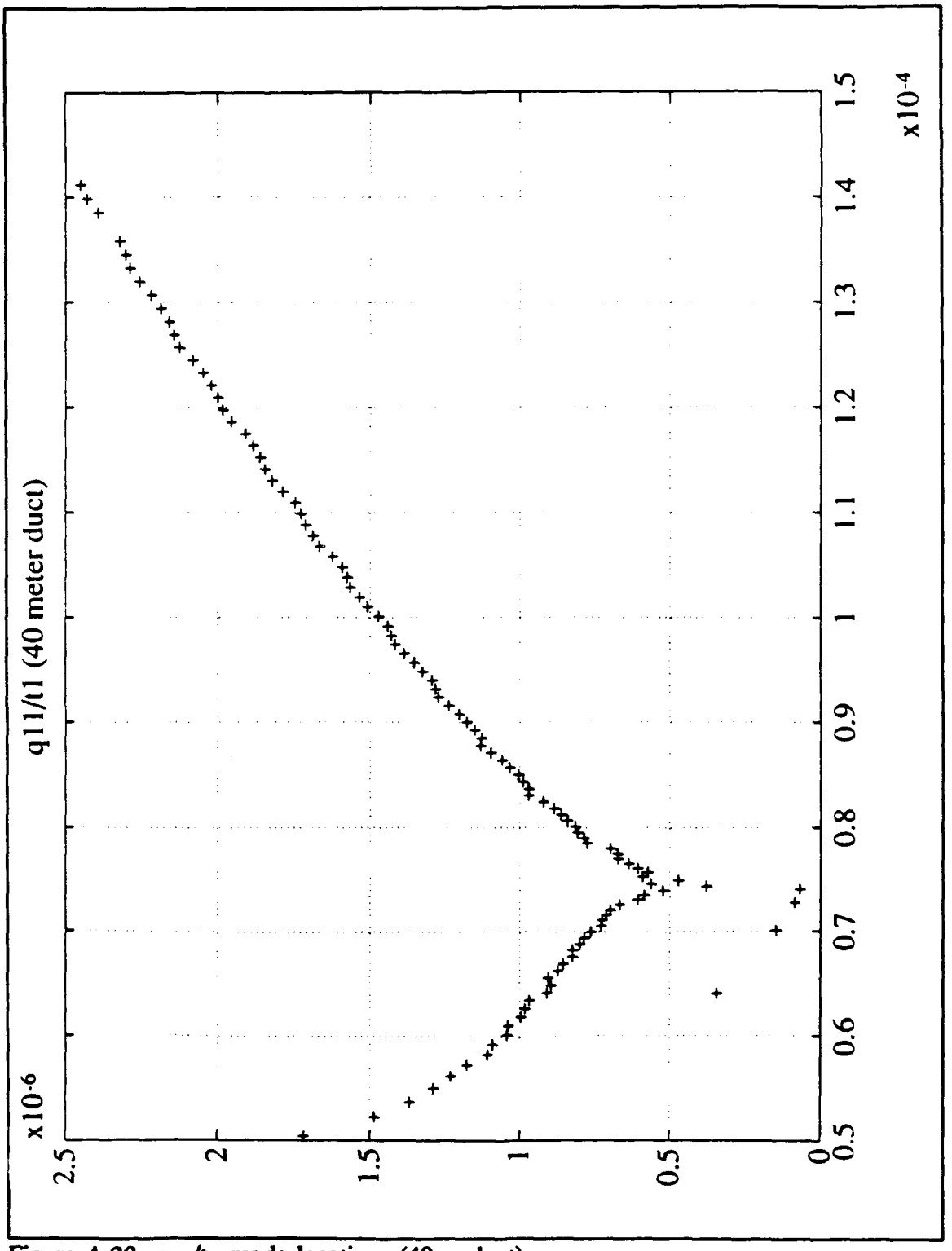


Figure A.20 q_{11}/t_1 mode locations (40 m duct).

APPENDIX B: SUBROUTINE FNDMOD AND FZEROX

This appendix contains the listing of the subroutines FNDMOD and FZEROX, modified from the NPS version of the M-Layer FORTRAN code to track the constant phase-lines and to locate the modes.

```
Line# Source Line     Microsoft FORTRAN Optimizing Compiler Version 5.00
   1       subroutine fndmod(aloss,dmmin,t1,dmesh,filem,nrmode,qeigen)
   2
   3  c purpose:
   4  c     This subroutine sets up the areas in the complex q11-plane
   5  c     to search for the zeroes of the modal function.
   6  c
   7  c inputs...
   8  c        mxlayr - maximum number of layers allowed in refractivity
   9  c                 profile
  10  c        nzlayr - actual number of layers in refractivity profile
  11  c        aloss  - maximum attenuation rate (in db/km) of modes
  12  c                 to be found
  13  c        dmmin  - minimum of zim(j)-zim(1)
  14  c        t1     - koa123
  15  c        dmesh  - initial mesh size divisor
  16  c
  17  c outputs...
  18  c        qeigen - complex array containing the zeros of the modal
  19  c                 function
  20  c        nrmode - actual number of modes found
  21  c
  22  c subroutines called...
  23  c        fzerox
  24  c        findfx
  25  c common block areas...
  26  c        com1
  27  c**********
  28  c
  29        implicit double precision (a-h,o-z)
  30        complex*16 ctemp,t1,qeigen,zeros,fx1,fx2
  31        parameter(c20log=8.68588963806504d0,one=(1.d0,0.d0),
  32       $          step0=1024.d0,step0h=step0/2.d0)
  33         character*3   filem,kline
  34         character*40  rfile
  35  c
  36  c**********
  37  c     qeigen    - complex array containing all the zeros of the
  38  c                 modal function found
  39  c     zeros     - complex array containing the zeros of the modal
  40  c                 function found in the current rectangular region
  41  c                 of the complex q11-plane
  42  c
  43  c**********
  44  c       use include file for parameters of
  45  c              mxlayr  max # layers
  46  c              mxmode  max # modes
  47  c
  48  $include: 'mlaparm.inc'
  ***** Begin listing of: mlaparm.inc
```

```
1  c
2  c       include file to define the
3  c               maximum # of layers (mxlayr)
4  c               maximum # of modes  (mxmode)
5  c
6         parameter (mxlayr=35 )
7         parameter (mxmode=127 )
 ***** End listing of: mlaparm.inc
49
50        dimension qeigen(mxmode),zeros(2*mxmode+1)
51  c*****
52        common /com1/waveno
53  c
54  c*************************************************************************
55  c     r1    - value of mode search on the real part of q11 at the
56  c             left edge in tmesh units.
57  c     r2    - value of mode search on the real part of q11 at the
58  c             right edge in tmesh units.
59  c     bot   - value of the imaginary part of q11 at the bottom
60  c             edge (this is set to 0).
61  c     t0    - value of the imaginary part of q11 at the top edge
62  c             in tmesh units.
63  c     step  - size of search areas.
64  c*************************************************************************
65  c
66  c
67  c ---------------------------------------------------------------
68  c                   start of executable statements
69  c ---------------------------------------------------------------
70  c*************************************************************************
71  c     set up search areas for finding modes and solve for modes,
72  c     calculate approximate value for re(q11) where modes start
73  c     occurring
74  c*************************************************************************
75  c
76        nrmode=0
77  c
78        recons=dreal(t1)
79        rstart=-2.0d-6*dmmin
80        qtest=rstart*recons
81
82        ttop1=(2.0d-3/(waveno*c20log))*recons
83        ttop=aloss*ttop1
84        tmesh=ttop1/dmesh
85        if(tmesh .gt. 0.1d0) tmesh=1.0d-1
86        if(tmesh .gt. 1.0d-3) then
87          tol=1.0d-4
88        else
89          tol=tmesh*0.1d0
90        end if
```

```
c
c*****
      write(*,1002)
      write(16,1002)
c
      write(*,1003) tmesh,tol
      write(16,1003) tmesh,tol
c
      t0=dnint(ttop/tmesh)+1.d0
      bot=0.d0
c*****
c
      r1=dnint(qtest/tmesh)-step0h
      rl=r1
      rr=r1+step0
      call findfx(r1,t0,fx1,x1,y1,tmesh)
      i1=int(r1)
      in=int(step0)+int(r1)
      nline=0
 50    continue
      do 100 nn=i1,in
        r2=r1+1.d0
        call findfx(r2,t0,fx2,x2,y2,tmesh)
        if (y1*y2 .lt. 0.d0) then
         r=r1*tmesh
         write (*,5555) r
c         write (16,5555) r
5555      format (/5x,'r= ',d28.16)
         go to 400
        endif
        r1=r2
        fx1=fx2
        x1=x2
        y1=y2
 100   continue
      return
c
 400    continue
       nrold=nrmode
       nline=nline+1
        nc100=int(nline/100)
        nc10=int((nline-nc100*100)/10)
        nc1=nline-nc100*100-nc10*10
       kline=char(48+nc100)//char(48+nc10)//char(48+nc1)
       rfile='e:\ans5\r'//filem//kline//'.dat'
c******
      open (unit=25,file=rfile,status='unknown')

      call fzerox(t0,r1,r11,fx1,x1,y1,fx2,x2,y2,zeros,nrold,nrnew,
     $            tmesh)
```

```
      close (25)
c******
c
      nrmode=nrnew
      new0=nrmode-nrold
c
c*************************************************************************
c     mode search completed
c     order zeros found by order of increasing real part.
c*************************************************************************
c
      if(nrmode .gt. 1) then
        jkend=nrmode-1

       do 420 jk=1,jkend
          nend=nrmode-jk

          do 410 ja=1,nend
            nrj=nrmode-ja
            nrj1=nrj+1
            if(dimag(zeros(nrj1)).lt.
     $         dimag(zeros(nrj))) then
              ctemp=zeros(nrj1)
              zeros(nrj1)=zeros(nrj)
              zeros(nrj)=ctemp
          end if
410       continue
420     continue
      end if
c
c*************************************************************************
c   the possibility exists that duplicate (within the tolerance 'tol')
c   zeros of the modal equation will be found.  eliminate these
c   duplicate zeros.
c*************************************************************************
c
      jkflag=0
      jkend=nrmode-1

      do 240 jk=1,jkend
        jk1=jk+1
        ctemp=zeros(jk1)
       chksq=cdabs(zeros(jk-jkflag)-ctemp)
       if(chksq .lt. tol) then
         jkflag=jkflag+1
           go to 240
       end if
        zeros(jk1-jkflag)=ctemp
240   continue
      nrmode=nrmode-jkflag
```

```
c
c*********************************************
c  Store the zeros as the eigenvalues.
c*********************************************
c
      nrmode=min0(nrmode,mxmode)

      do 430 jk=1,nrmode
       qeigen(jk)=zeros(jk)
430   continue
c
      i1=int(r11)+1
      if (i1 .lt. in) then
       r1=r11+1.d0
       call findfx(r1,t0,fx1,x1,y1,tmesh)
       go to 50
      endif
      return
c
c-------------------------------------------------------------------
c                     format statements
c-------------------------------------------------------------------
1000  format(/5x,'searching for zeroes in this areas are',
     $   ' defined by:'/5x,'istart= ',i10/5x,'itop= ',i10,
     $   5x,' ibot= ',i11/5x,'ileft= ',i10,5x,'iright= ',i10)

1001  format(5x,i4,' new zeroes found in this area.'/)

1002  format(//5x,'*******      start search for modal eigenvalues',
     $      '     *******')

1003  format(/5x,'tmesh= ',d15.5,5x,'tol= ',d15.5/)

      end
c
c
c*************************************************************************
c*************************************************************************
c
      subroutine fzerox(t0,r1,r11,fx1,x1,y1,fx2,x2,y2,zeros,nrold,
     $                  nrnew,tmsh0)
c
c*****
c  fzerox is a routine for finding the zeroes of a complex function, f,
c     which lie within a specified rectangular region of the
c     complex q11 plane, assuming that the function has only
c     simple zeroes over this rectangle.
c
c  parameters specifying the search rectangle:
c     tmesh - set equal to about half the average spacing between
```

```
241 c            zeroes within the rectangle.  A smaller value may be used
242 c            as a safety measure, but too small a value will result
243 c            in excessively long run time.
244 c     zeros - output list of (complex) values of q11 at which
245 c            zeroes are found.
246 c  nrnew,nrold- the number of zeroes found
247 c
248 c  subroutines called--
249 c        findfx
250 c        roots
251 c*****
252       implicit double precision (a-h,o-z)
253       complex*16 f10,f01,f11,fx1,fx2,fx00,fx10,fx01,fx11,
254      +           one,sol,zeros
255       parameter(one=(1.d0,0.d0))
256  $include: 'mlaparm.inc'
  ***** Begin listing of: mlaparm.inc
 1 c
 2 c      include file to define the
 3 c            maximum # of layers (mxlayr)
 4 c            maximum # of modes  (mxmode)
 5 c
 6       parameter (mxlayr=35 )
 7       parameter (mxmode=127 )
  ***** End listing of: mlaparm.inc
257       dimension sol(3),theta(2),zeros(2*mxmode+1),rline(2,1024)
258 c
259       common/tmccom/tmesh
260 c*****
261 c
262       npoint=0
263       nrnew=nrold
264       tmesh=tmsh0
265       bot=0.d0
266       t=t0-1.d0
267       r11=r1
268       r=r11
269 c*****
270 c
271       fx01=fx1
272       fx01r=x1
273       fx01i=y1
274       fx11=fx2
275       fx11r=x2
276       fx11i=y2
277       go to 82
278 c
279 c*****
280 c  enter mesh square from left side or exit rectangle at right edge.
281 c
```

```
 20    r=r+1.d0
 21    fx01=fx11
       fx01r=fx11r
       fx01i=fx11i
       fx00=fx10
       fx00r=fx10r
       fx00i=fx10i
 22    continue
       call findfx(r+1.d0,t+1.d0,fx11,fx11r,fx11i,tmesh)
       call findfx(r+1.d0,t,fx10,fx10r,fx10i,tmesh)
c*******
c     Determine the edge of exit of im(f)=0 from current mesh.
       edgeit=fx01i*fx11i
       edgeib=fx00i*fx10i
       if (edgeib .gt. 0.d0) then
c        Im(f)=0 goes through the 01 to 10 line.
         if (edgeit .gt. 0.d0) then
c        Im(f)=0 goes through the 10 to 11 edge (edge 1).
           lout=1
         else
c        Im(f)=0 goes through the 01 to 11 edge (edge 2)
           lout=2
         end if
       else
c        Im(f)=0 goes through the 00 to 10 edge (edge 4)
         lout=4
         if (edgeit .lt. 0.d0) then
c        Im(f)=0 also runs through 01 to 11 and 10 to 11 edges.
c        Store crossing location and in/out information.
           knot34=knot34+1
c          loc34r(knot34)=lr
c          loc34i(knot34)=li
         end if
       end if
c*******
       go to 85
c*****
c  enter mesh square from bottom side or exit rectangle at top edge.
 40    t=t+1.d0
 41    fx00=fx01
       fx00r=fx01r
       fx00i=fx01i
       fx10=fx11
       fx10r=fx11r
       fx10i=fx11i
 42    continue
       call findfx(r,t+1.d0,fx01,fx01r,fx01i,tmesh)
       call findfx(r+1.d0,t+1.d0,fx11,fx11r,fx11i,tmesh)
c*******
c     Determine the edge of exit of im(f)=0 from current mesh.
```

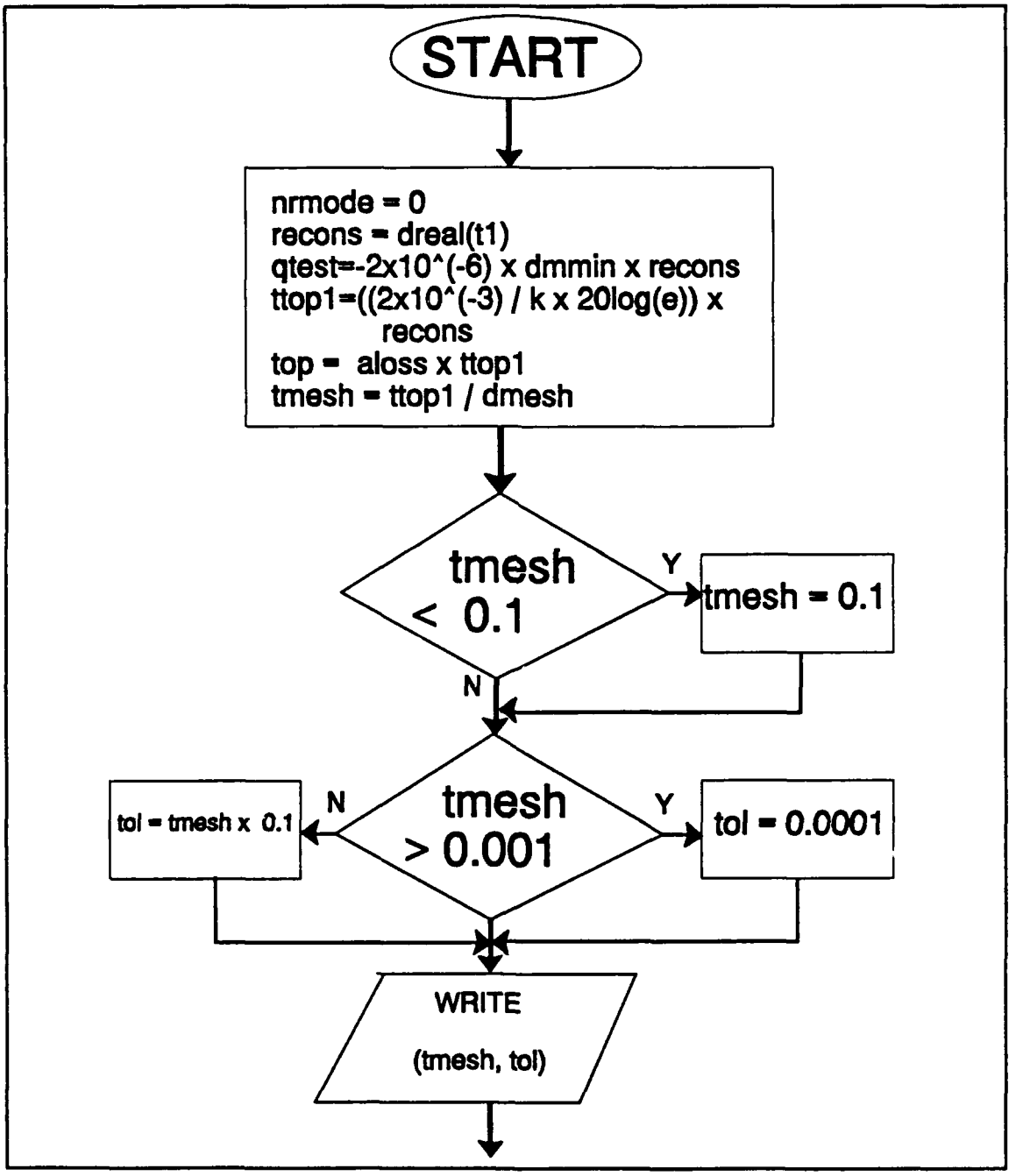
```
      edgeil=fx00i*fx01i
      edgeir=fx10i*fx11i
      if (edgeir .gt. 0.d0) then
c       Im(f)=0 goes through the 00 to 11 line.
        if (edgeil .gt. 0.d0) then
c         Im(f)=0 goes through the 01 to 11 edge (edge 2)
          lout=2
        else
c       Im(f)=0 goes through the 00 to 01 edge (edge 3).
          lout=3
        end if
      else
c       Im(f)=0 goes through the 10 to 11 edge (edge 1)
        lout=1
        if (edgeil .lt. 0.d0) then
c       Im(f)=0 also runs through 00 to 01 and 01 to 11 edges.
c       Store crossing location and in/out information.
          knot41=knot41+1
c         loc41r(knot41)=lr
c         loc41i(knot41)=li
        end if
      end if
c*******
      go to 85
c*****
c  enter mesh square from right side or exit rectangle at left edge.

60    r=r-1.d0
61    fx11=fx01
      fx11r=fx01r
      fx11i=fx01i
      fx10=fx00
      fx10r=fx00r
      fx10i=fx00i
65    continue
      call findfx(r,t+1.d0,fx01,fx01r,fx01i,tmesh)
      call findfx(r,t,fx00,fx00r,fx00i,tmesh)
c*******
c     Determine the edge of exit of im(f)=0 from current mesh.
      edgeit=fx01i*fx11i
      edgeib=fx00i*fx10i
      if (edgeit .gt. 0.d0) then
c       Im(f)=0 goes through the 01 to 10 line.
        if (edgeib .gt. 0.d0) then
c       Im(f)=0 goes through the 00 to 01 edge (edge 3).
          lout=3
        else
c       Im(f)=0 goes through the 00 to 10 edge (edge 4)
          lout=4
        end if
```

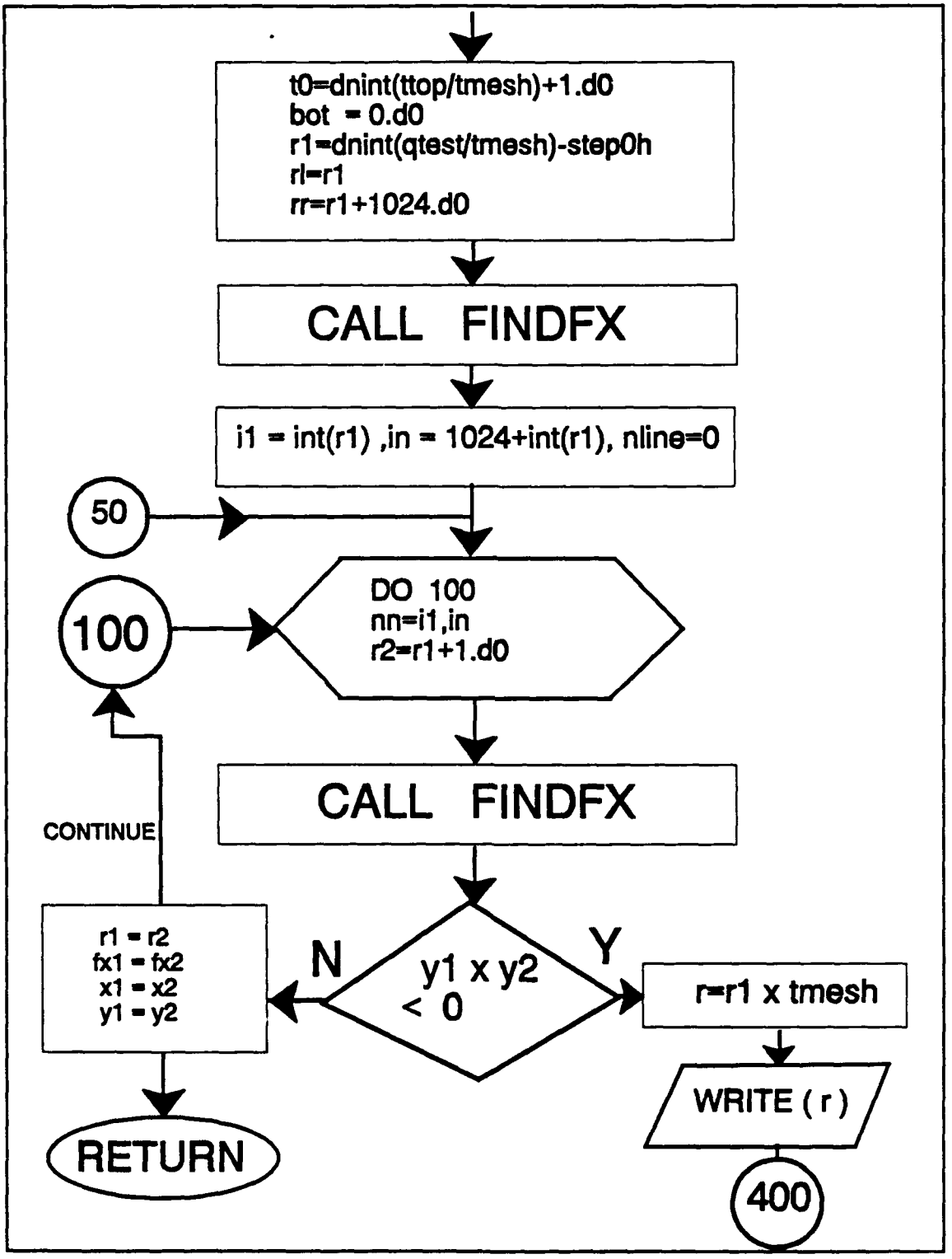
```
382       else
383  c      Im(f)=0 goes through the 01 to 11 edge (edge 2)
384         lout=2
385         if (edgeib .lt. 0.d0) then
386  c      Im(f)=0 also runs through 00 to 10 and 00 to 01 edges.
387  c      Store crossing location and in/out information.
388           knot12=knot12+1
389  c        loc12r(knot12)=lr
390  c        loc12i(knot12)=li
391         end if
392       end if
393  c*******
394       go to 85
395  c*****
396  c enter mesh square top side or exit rectangle at bottom edge.
397  c
398  80    t=t-1.d0
399  81    fx01=fx00
400       fx01r=fx00r
401       fx01i=fx00i
402       fx11=fx10
403       fx11r=fx10r
404       fx11i=fx10i
405  82    continue
406       call findfx(r,t,fx00,fx00r,fx00i,tmesh)
407       call findfx(r+1.d0,t,fx10,fx10r,fx10i,tmesh)
408  c
409  c*******
410  c     Determine the edge of exit of im(f)=0 from current mesh.
411  c
412  83    continue
413       edgeil=fx00i*fx01i
414       edgeir=fx10i*fx11i
415       if (edgeil .gt. 0.d0) then
416  c      Im(f)=0 goes through the 00 to 11 line.
417         if (edgeir .gt. 0.d0) then
418  c        Im(f)=0 goes through the 00 to 10 edge (edge 4)
419           lout=4
420         else
421  c      Im(f)=0 goes through the 10 to 11 edge (edge 1).
422           lout=1
423         end if
424       else
425  c      Im(f)=0 goes through the 00 to 01 edge (edge 3)
426         lout=3
427         if (edgeir .lt. 0.d0) then
428  c      Im(f)=0 also runs through 00 to 10 and 10 to 11 edges.
429  c      Store crossing location and in/out information.
430           knot23=knot23+1
431  c        loc23r(knot23)=lr
```

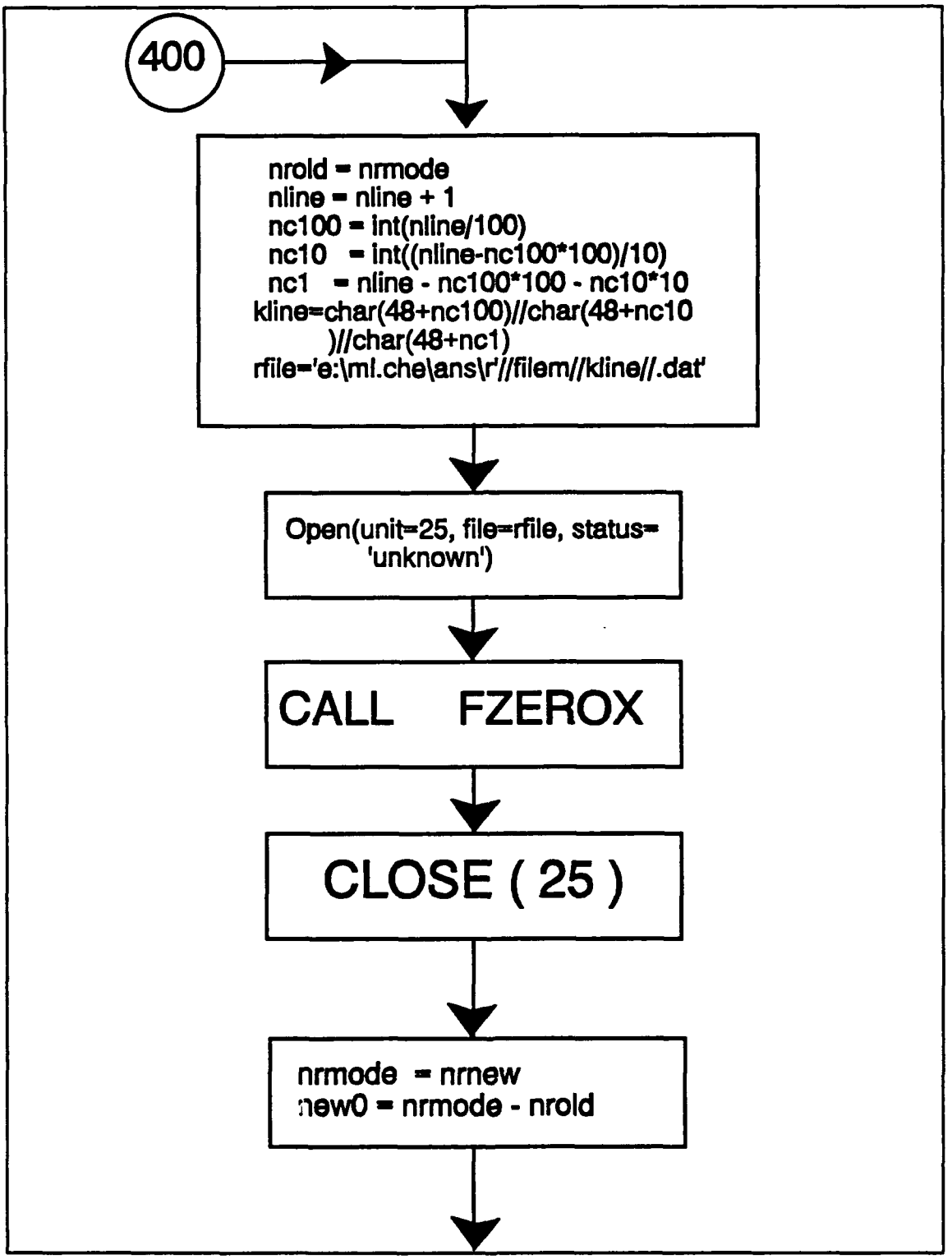
```
c          loc23i(knot23)=li
          end if
        end if
c*************************************************************************
c    Test for there being at least one re(f)=0 line entering and
c    leaving the mesh square.
c*************************************************************************
c
 85    continue
       npoint=npoint+1
       rrr=r*tmesh
       ttt=t*tmesh
       rline(1,npoint)=rrr
       rline(2,npoint)=ttt
       if (npoint .eq. 1024) lout=5
c
       if((t .lt. bot) .or. (t .gt. t0)) go to 100

       if ((fx00r*fx10r .gt. 0.d0) .and. (fx01r*fx11r .gt. 0.d0)
      +  .and. (fx00r*fx01r .gt. 0.d0)) go to (20,40,60,80,100)
      +  lout
c*************************************************************************
c    Computate the values of the modal function at the corners of
c    a mesh square to determine its Taylor series to the 3rd order
c    for estimating its root locations.
c*************************************************************************
c
c     f00=one
       f10=cdexp(fx10-fx00)-one
       f01=cdexp(fx01-fx00)-one
       f11=cdexp(fx11-fx00)-one
c
c********** estimate locations of zeroes by radicals*****************
c
       call roots(f10,f01,f11,sol,nrsol)
c
       do 90 n=1,nrsol
         ureal = dreal(sol(n))
         uimag = dimag(sol(n))
         if (ureal .lt. 0.d0 .or. ureal .gt. 1.0d0) go to 90
         if (uimag .lt. 0.d0 .or. uimag .gt. 1.0d0) go to 90
 92       theta(1)=(r+ureal)*tmesh
         theta(2)=(t+uimag)*tmesh
         nrnew = nrnew+1
         zeros(nrnew)=dcmplx(theta(1),theta(2))
 90    continue
c
c*****************************************
c     continue following the phase line
c*****************************************
```

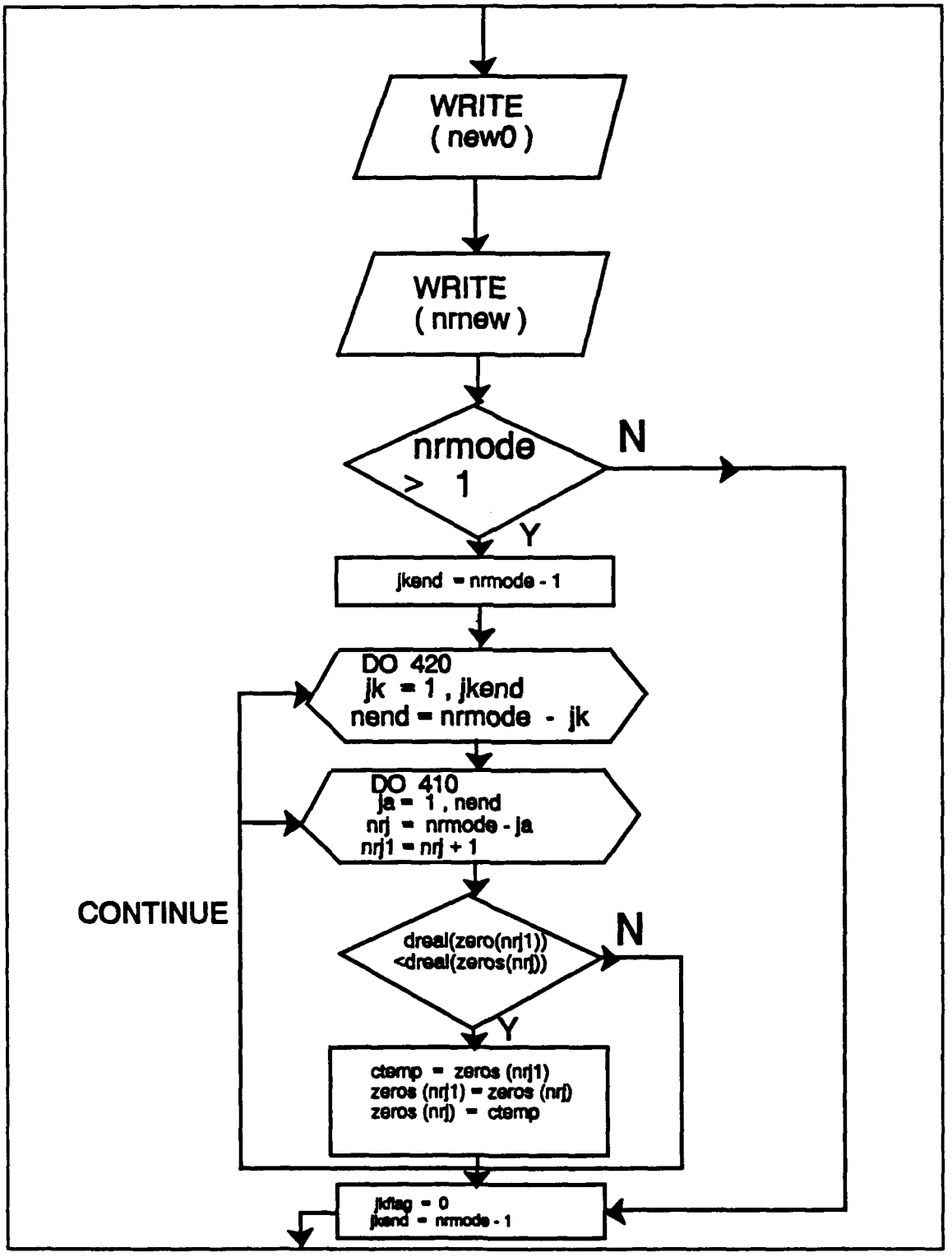
```
c
95    continue
      go to (20,40,60,80,100) lout
c**
100   continue
c******
      write (25,8888)  ((rline(i,j), i=1, 2), j=1, npoint)
8888  format (2e15.5)
c
      return
      end
```

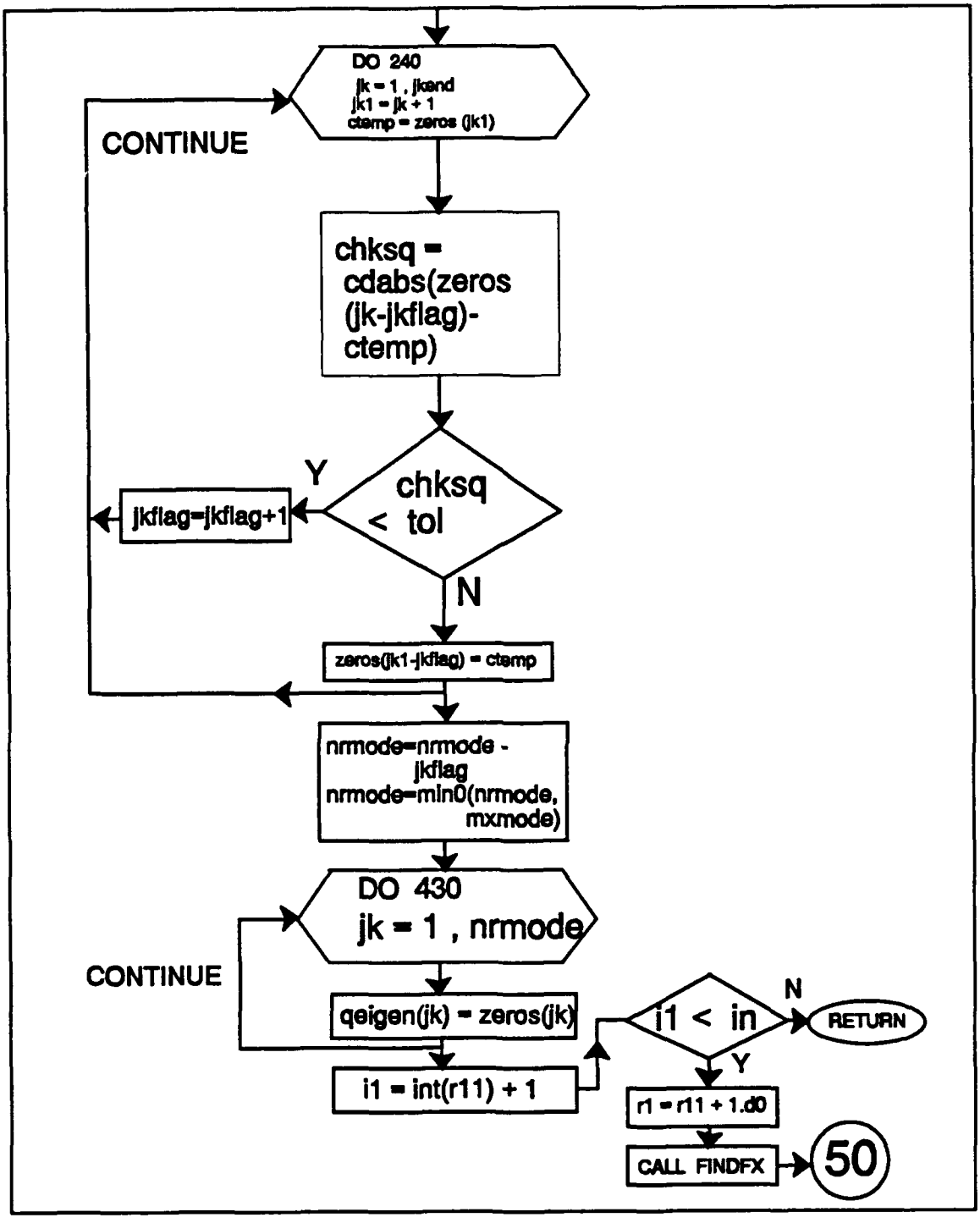
APPENDIX C: CONSTANT PHASE LINE TRACKING FLOW CHARTS




t0=dnint(ttop/tmesh)+1.d0
bot = 0.d0
r1=dnint(qtest/tmesh)-step0h
rl=r1
rr=r1+1024.d0
CALL FINDFX
i1 = int(r1) ,in = 1024+int(r1), nline=0
50
100
DO 100
nn=i1,in
r2=r1+1.d0
CALL FINDFX
CONTINUE
r1 = r2
fx1 = fx2
x1 = x2
y1 = y2
N
y1 x y2
< 0
Y
r=r1 x tmesh
WRITE (r)
400
RETURN


400
nrold = nrmode
nline = nline + 1
nc100 = int(nline/100)
nc10 = int((nline-nc100*100)/10)
nc1 = nline - nc100*100 - nc10*10
kline=char(48+nc100)//char(48+nc10
)//char(48+nc1)
rfile='e:\ml.che\ans\r'//filem//kline//.dat'
Open(unit=25, file=rfile, status=
'unknown')
CALL FZEROX
CLOSE (25)
nrmode = nrnew
new0 = nrmode - nrold


WRITE
(new0)
WRITE
(nrnew)
nrmode
> 1
N
Y
jkend = nrmode - 1
DO 420
jk = 1 , jkend
nend = nrmode - jk
DO 410
ja = 1 , nend
nrj = nrmode - ja
nrj1 = nrj + 1
CONTINUE
dreal(zero(nrj1))
<dreal(zeros(nrj))
N
Y
ctemp = zeros (nrj1)
zeros (nrj1) = zeros (nrj)
zeros (nrj) = ctemp
jkflag = 0
jkend = nrmode - 1


DO 240
jk = 1 , jkend
jk1 = jk + 1
ctemp = zeros (jk1)
CONTINUE
chksq = cdabs(zeros (jk-jkflag)- ctemp)
chksq < tol
Y
jkflag=jkflag+1
N
zeros(jk1-jkflag) = ctemp
nrmode=nrmode - jkflag
nrmode=min0(nrmode, mxmode)
DO 430
jk = 1 , nrmode
CONTINUE
qeigen(jk) = zeros(jk)
i1 = int(r11) + 1
i1 < in
N
RETURN
Y
r1 = r11 + 1.d0
CALL FINDFX
50

APPENDIX D: CONSTANT PHASE LINES IN THE COMPLEX q_{11}/t_1 PLANE

This appendix contains plots of the constant phase lines $\mathrm{Im}\{D(q)\} = 0$ and $\mathrm{Re}\{D(q)\} = 0$ initiating from the top edge of the search region in the complex q_{11}/t_1 plane for the evaporation ducts of 2, 4, 6, 8, 10, 20, 30, and 40 m heights.

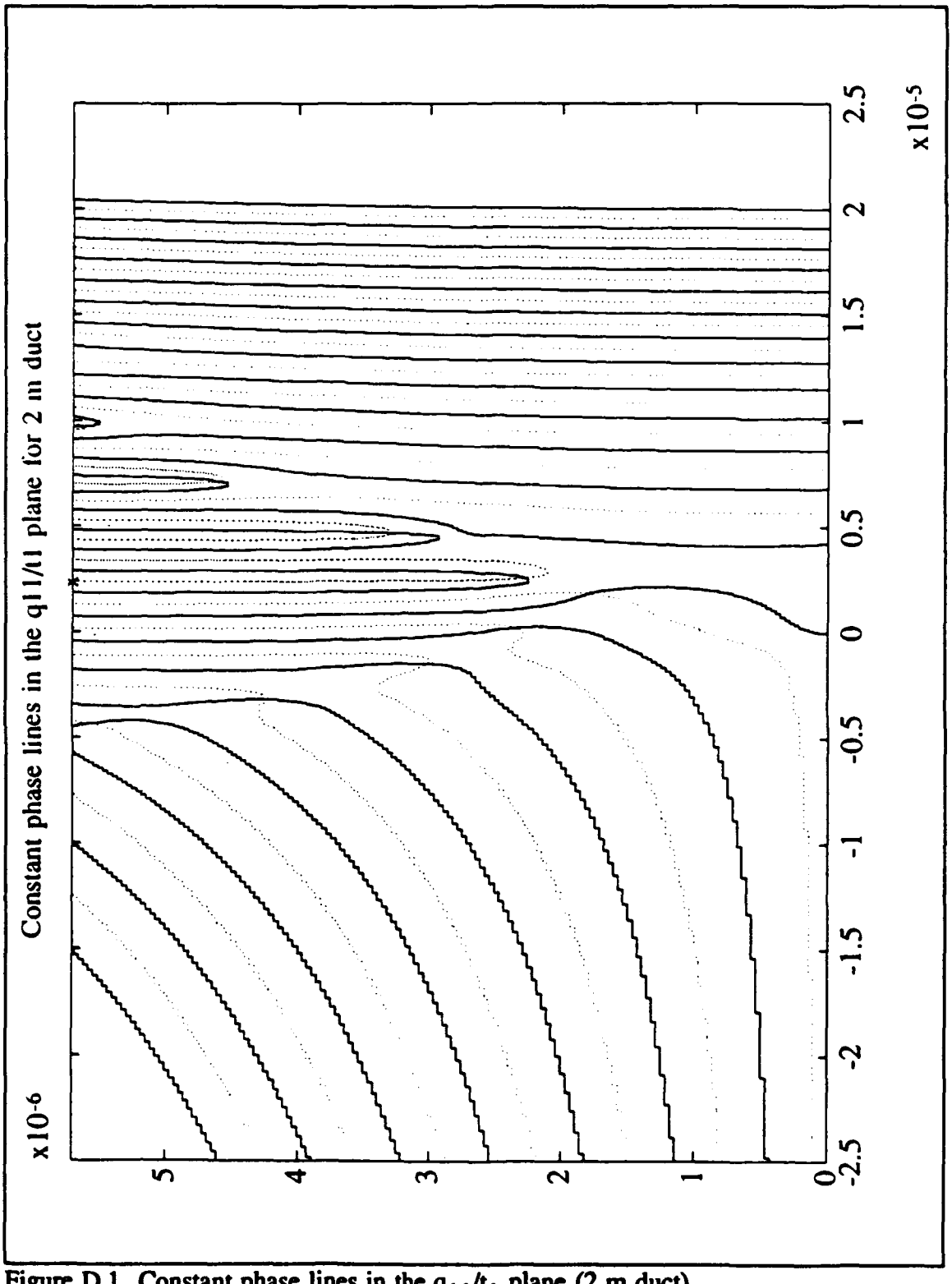


Figure D.1 Constant phase lines in the q_{11}/t_1 plane (2 m duct).

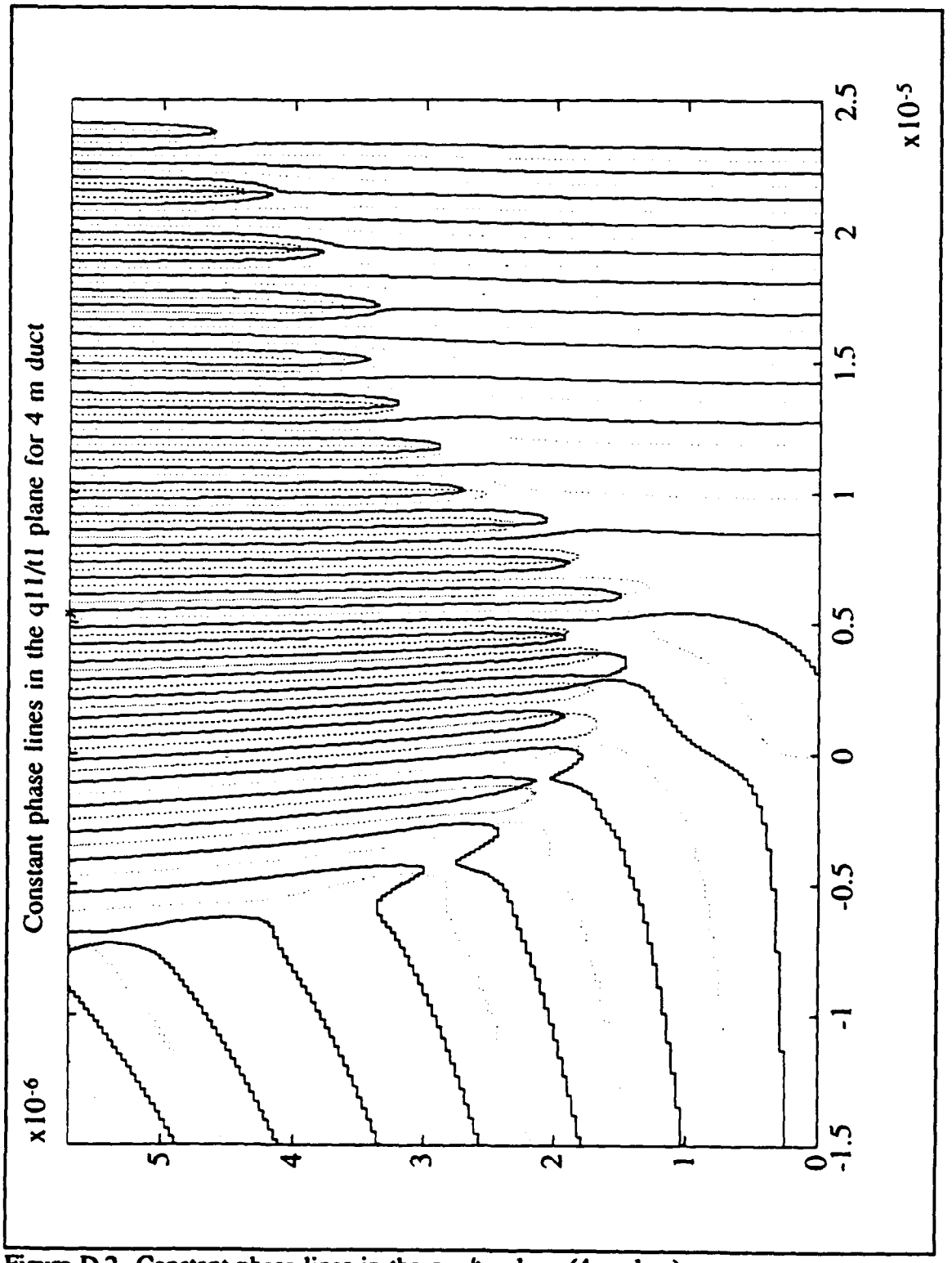


Figure D.2 Constant phase lines in the q_{11}/t_1 plane (4 m duct).

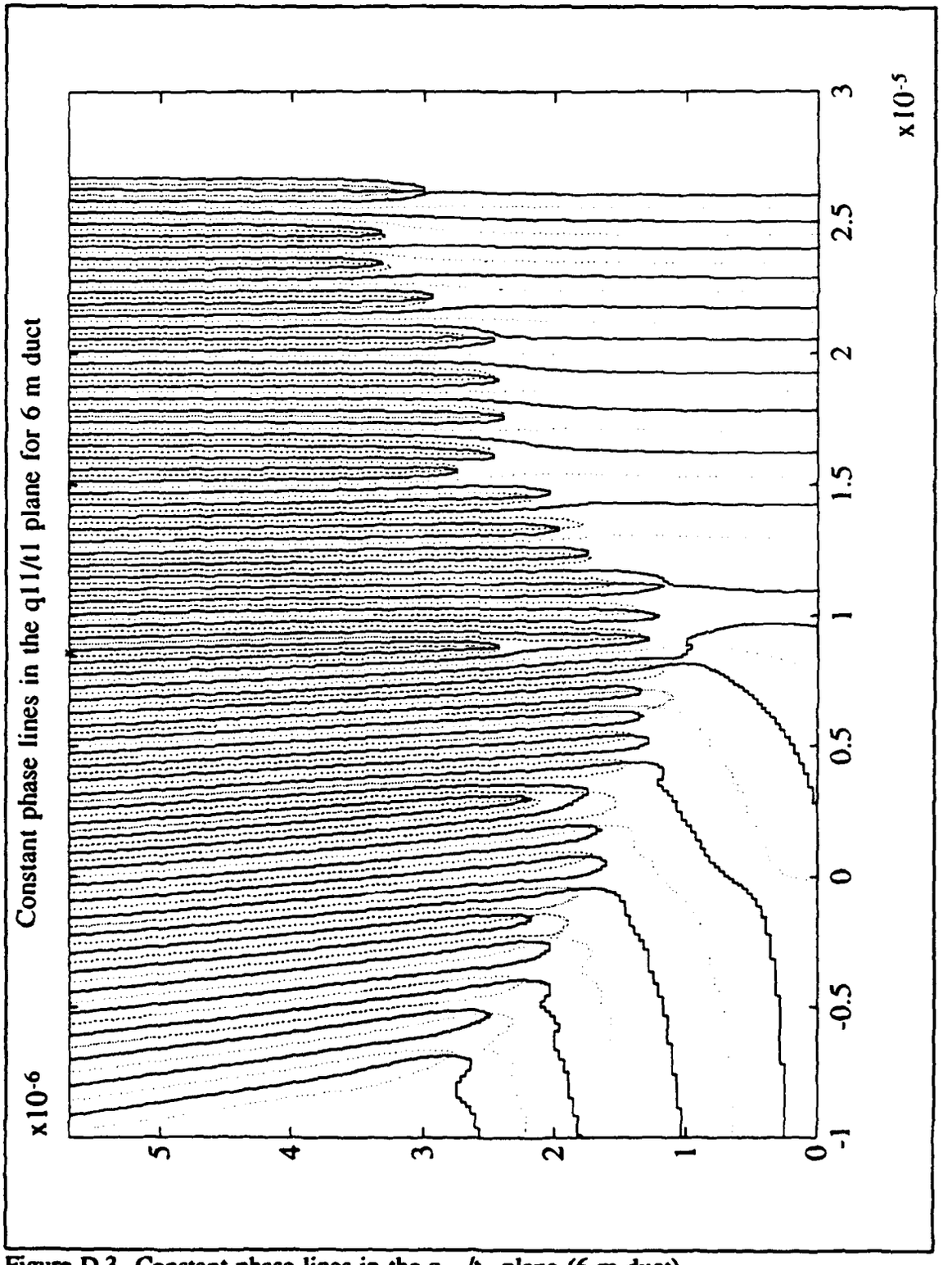


Figure D.3 Constant phase lines in the q_{11}/t_1 plane (6 m duct).

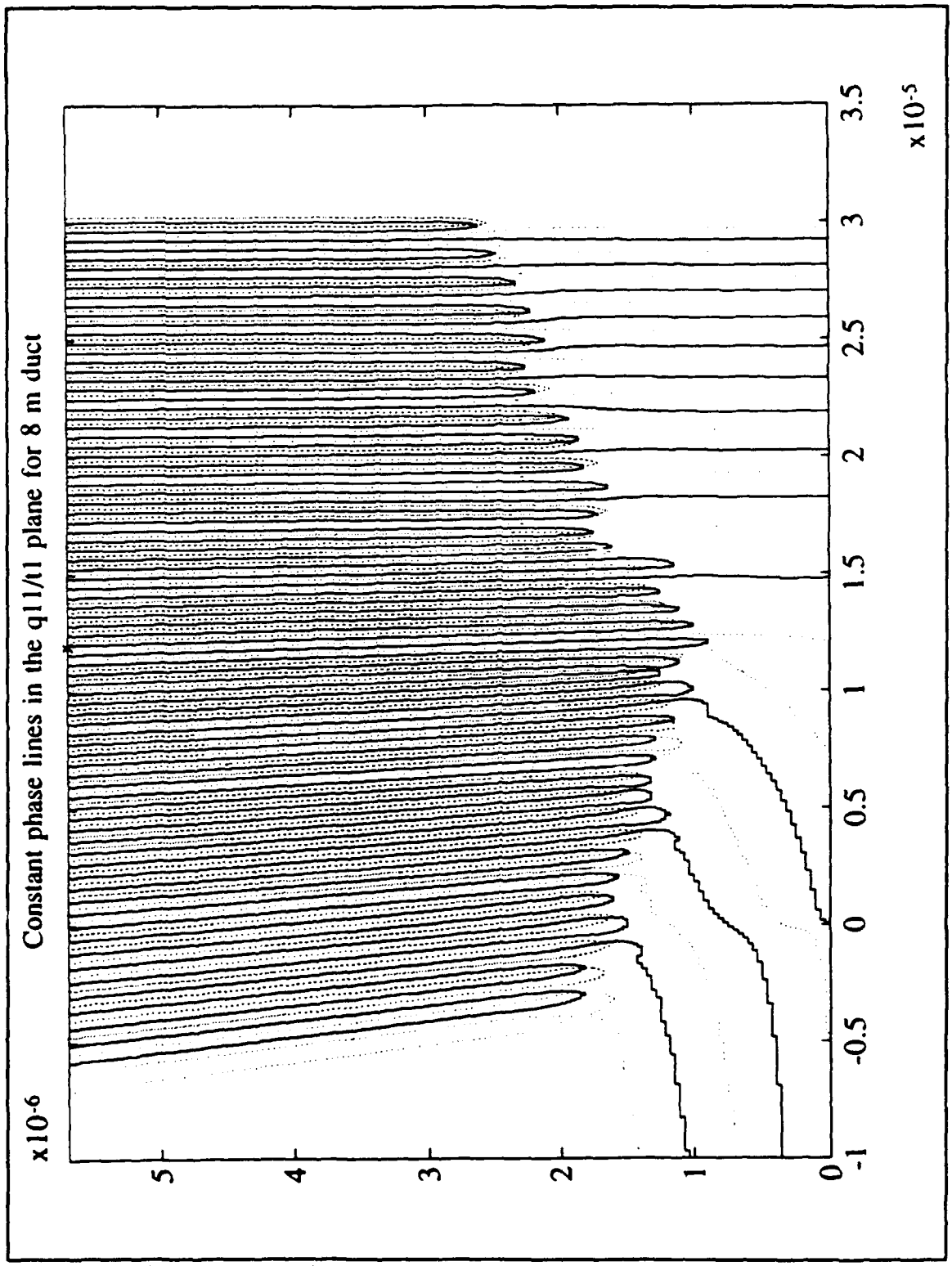


Figure D.4 Constant phase lines in the q_{11}/t_1 plane (8 m duct).

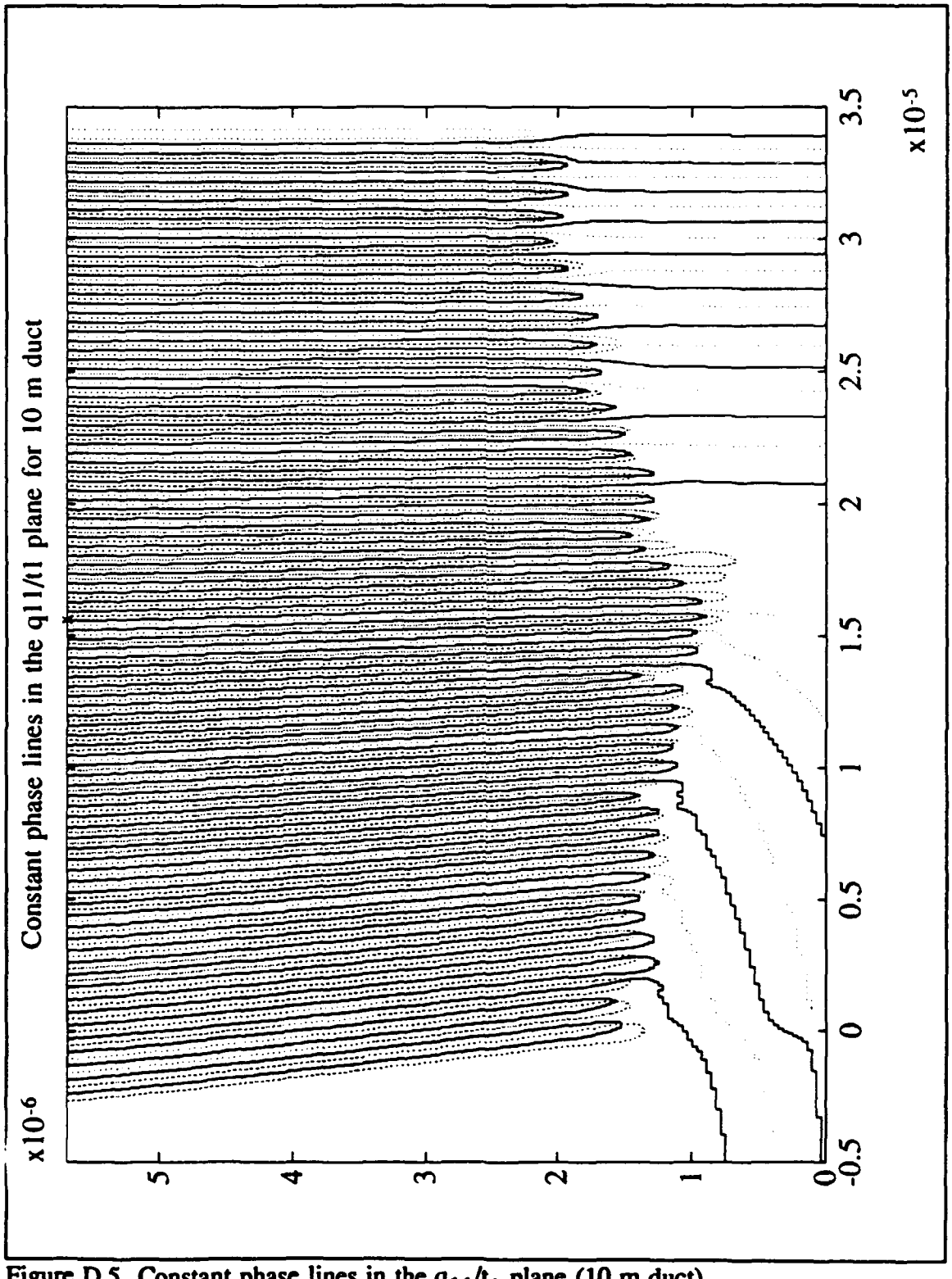


Figure D.5 Constant phase lines in the q_{11}/t_1 plane (10 m duct).

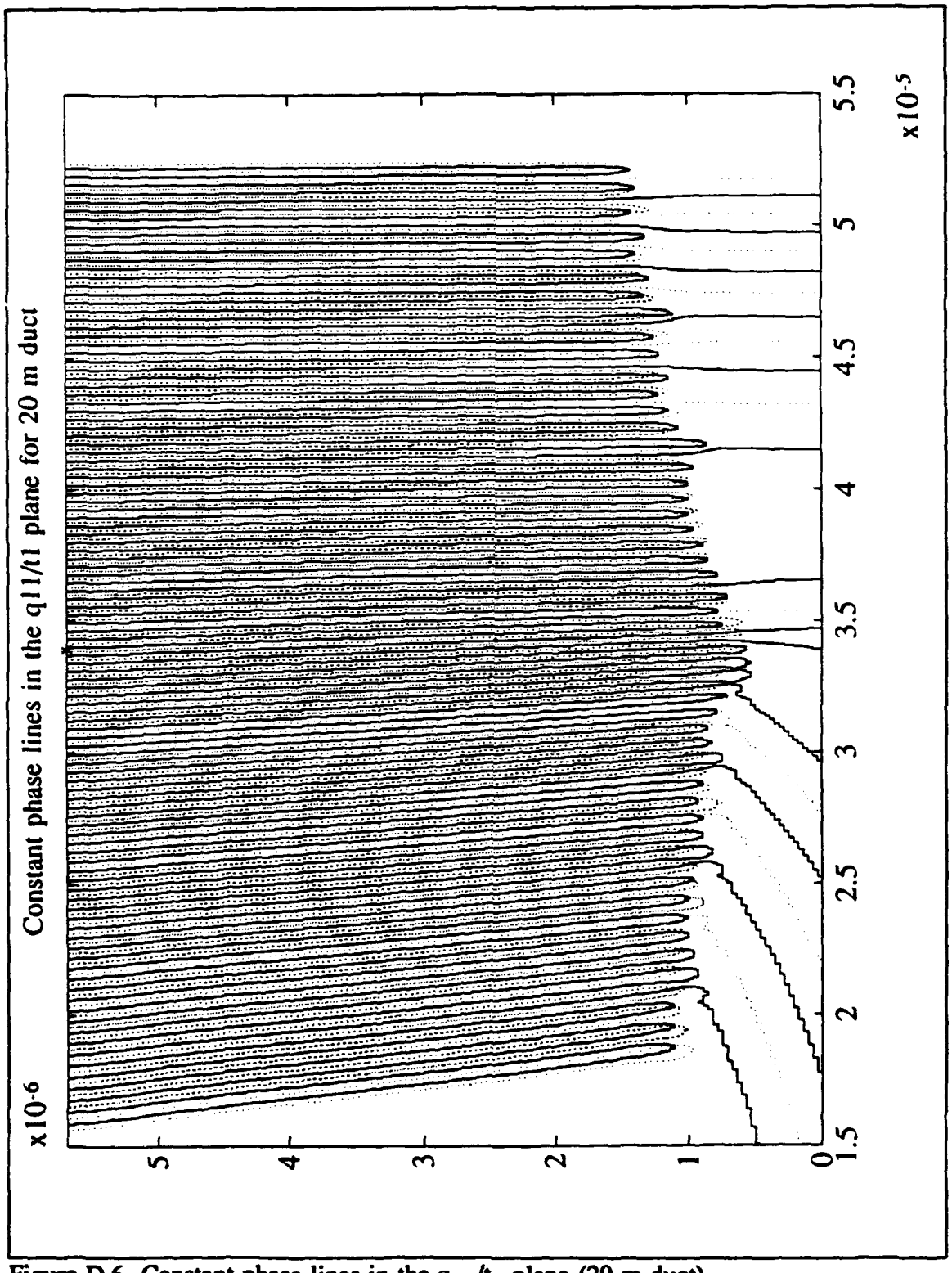


Figure D.6 Constant phase lines in the q_{11}/t_1 plane (20 m duct).

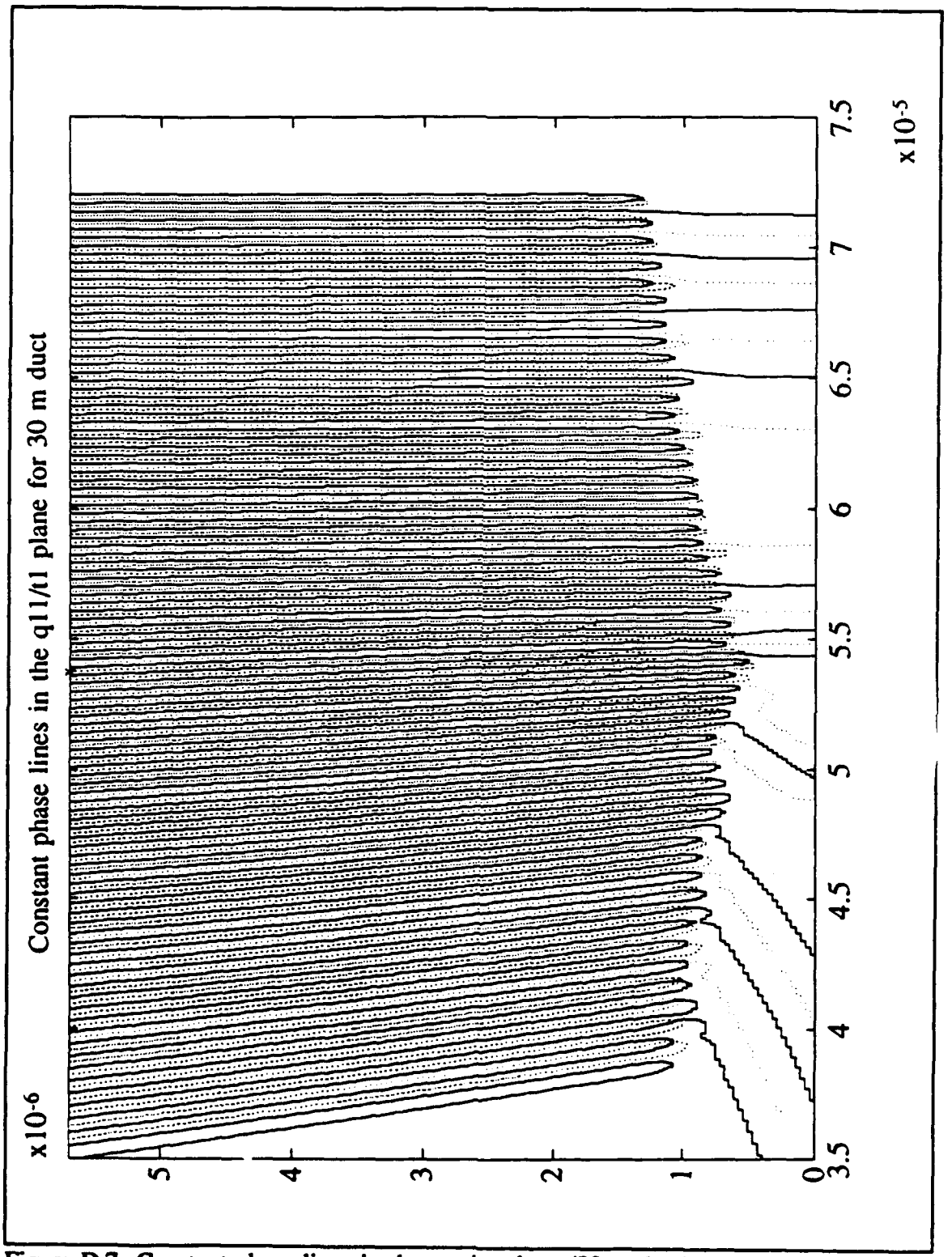


Figure D.7 Constant phase lines in the q_{11}/t_1 plane (30 m duct).

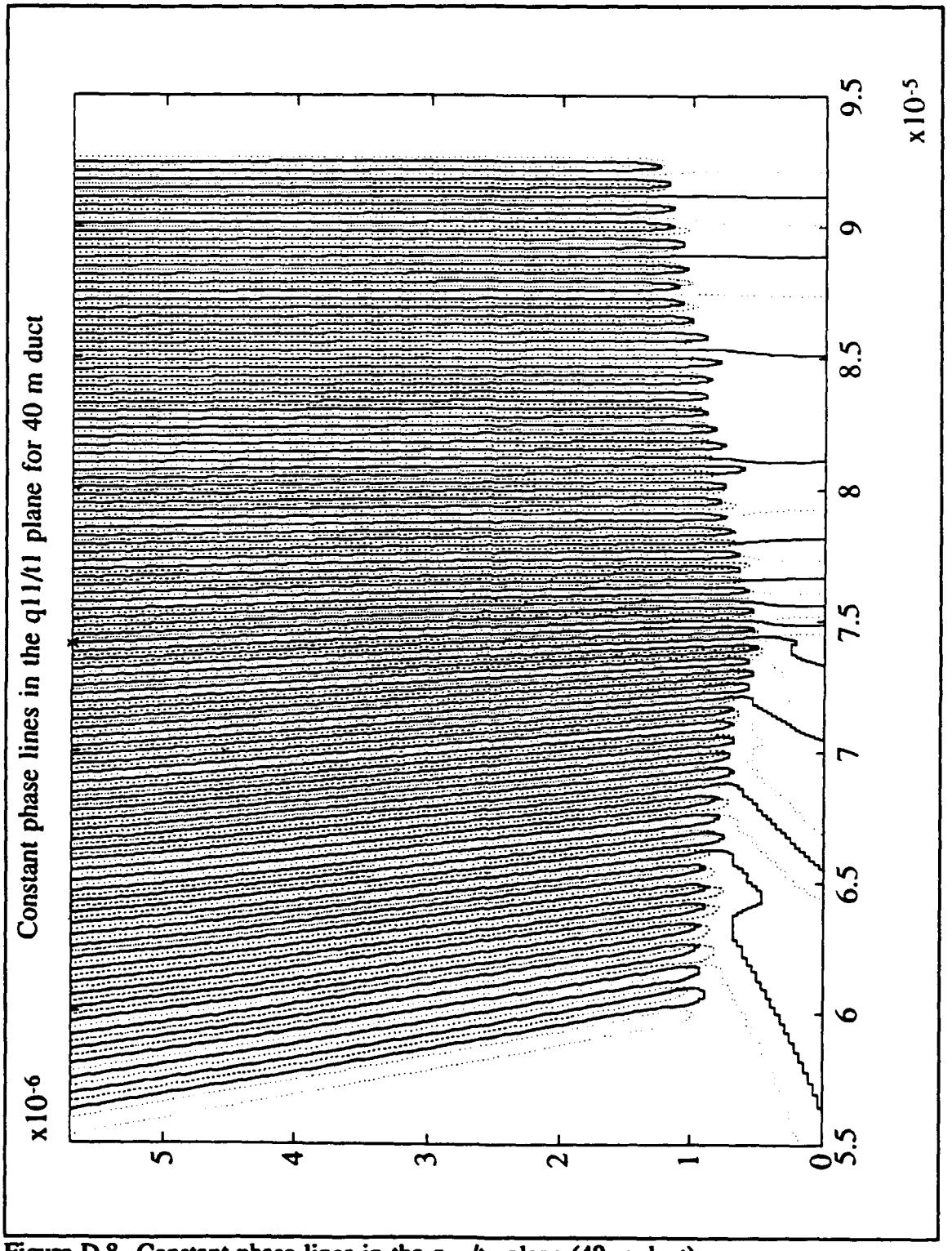


Figure D.8 Constant phase lines in the q_{11}/t_1 plane (40 m duct).

APPENDIX E: SEPARATION OF Im{D(q)}=0 LINES

This appendix contains figures of the separation of Im{D(q)}=0 lines along the top edge of the search region. The 2, 4, 6, 8, 10, 20, 30 and 40 meter evaporation ducts are included.

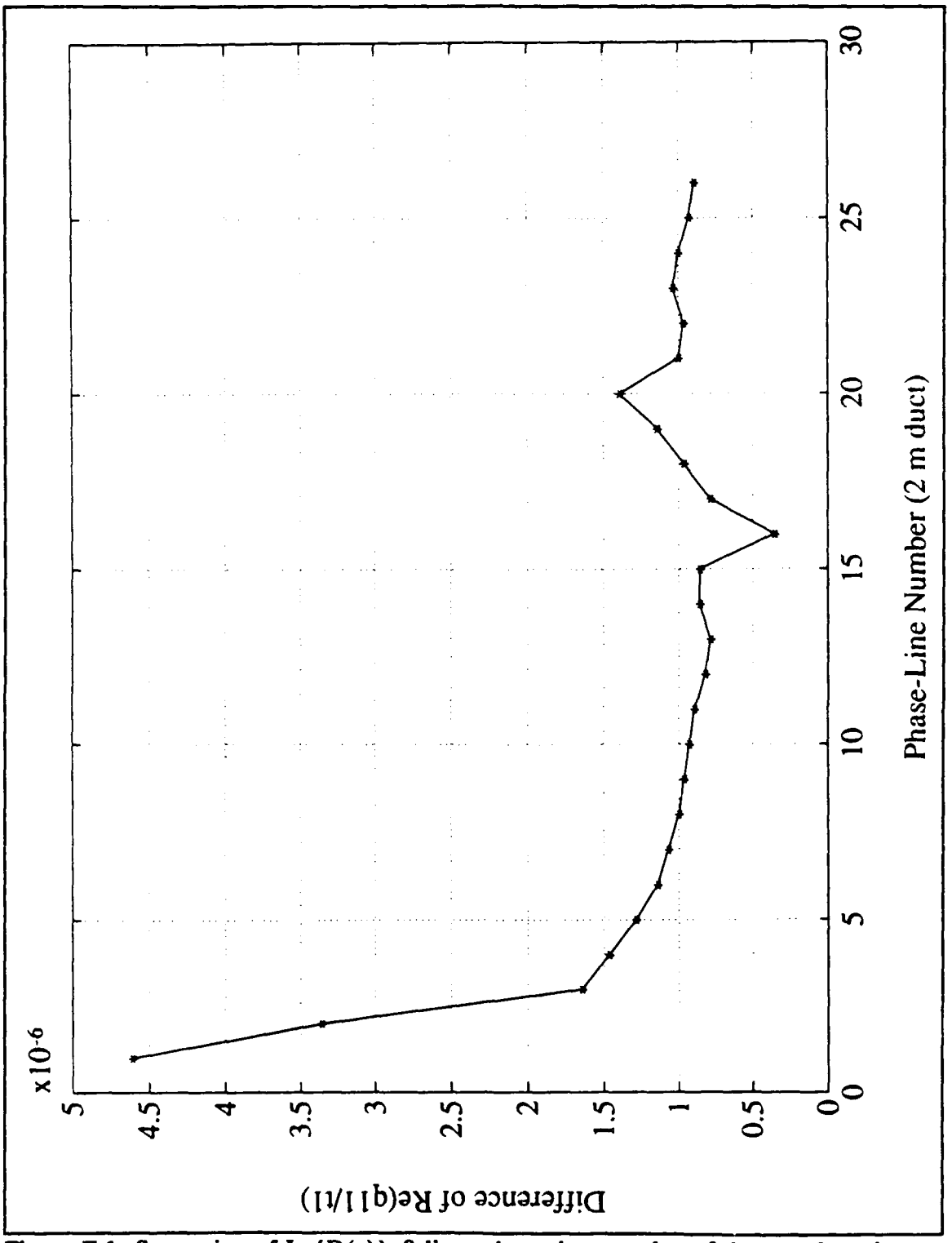


Figure E.1 Separation of Im{D(q)}=0 lines along the top edge of the search region (2 m duct).

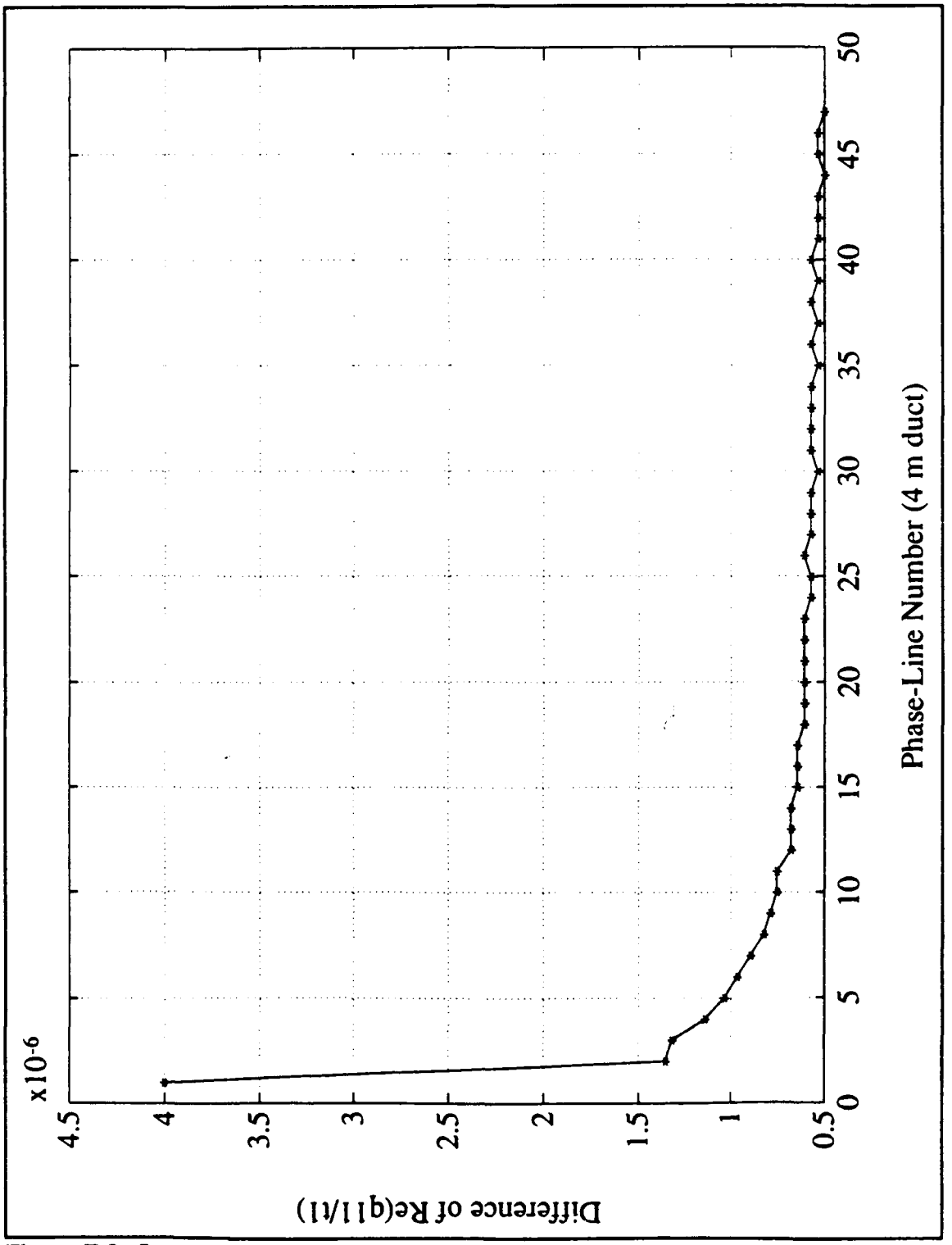


Figure E.2 Separation of Im{D(q)}=0 lines along the top edge of the search region (4 m duct).

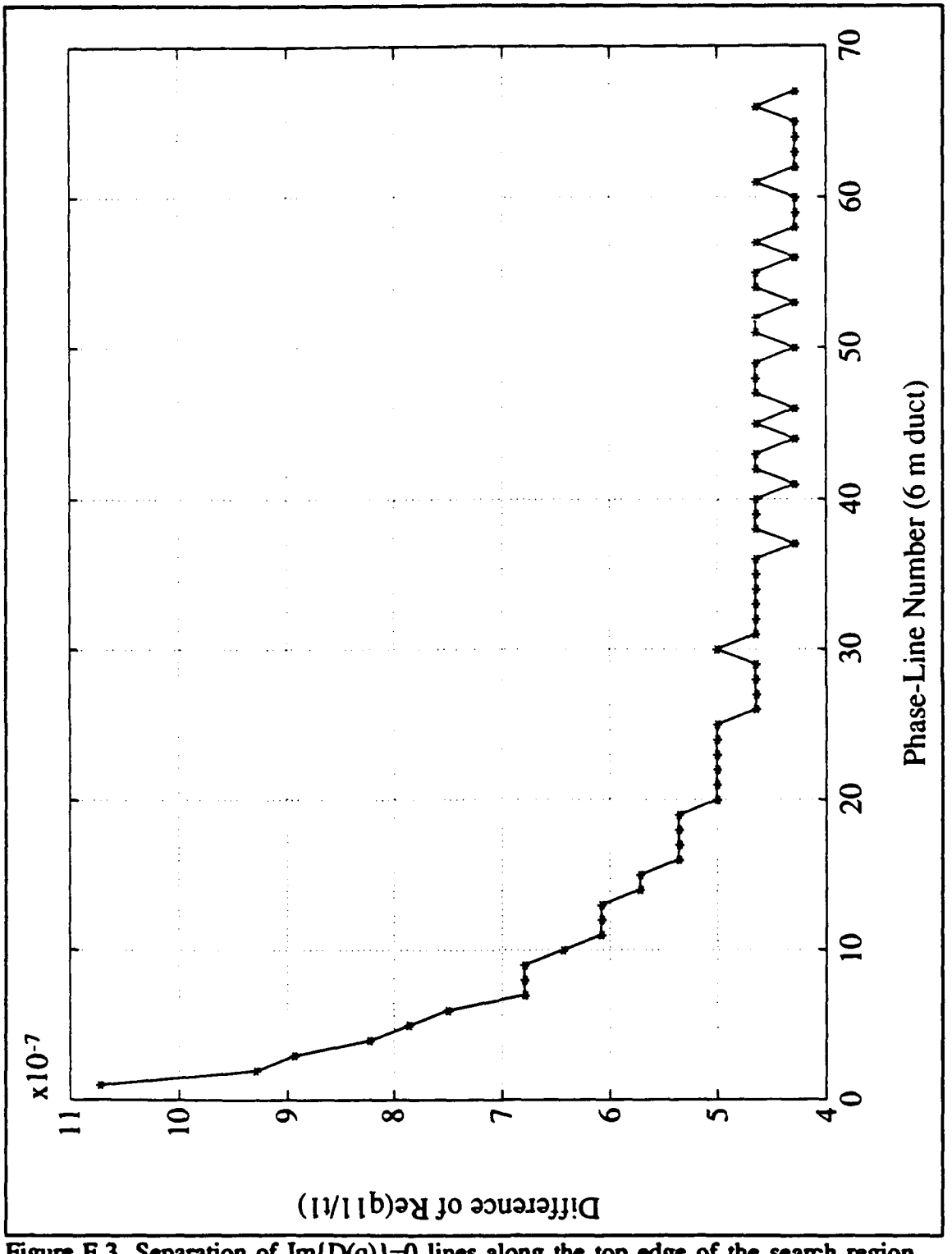


Figure E.3 Separation of Im{D(q)}=0 lines along the top edge of the search region (6 m duct).

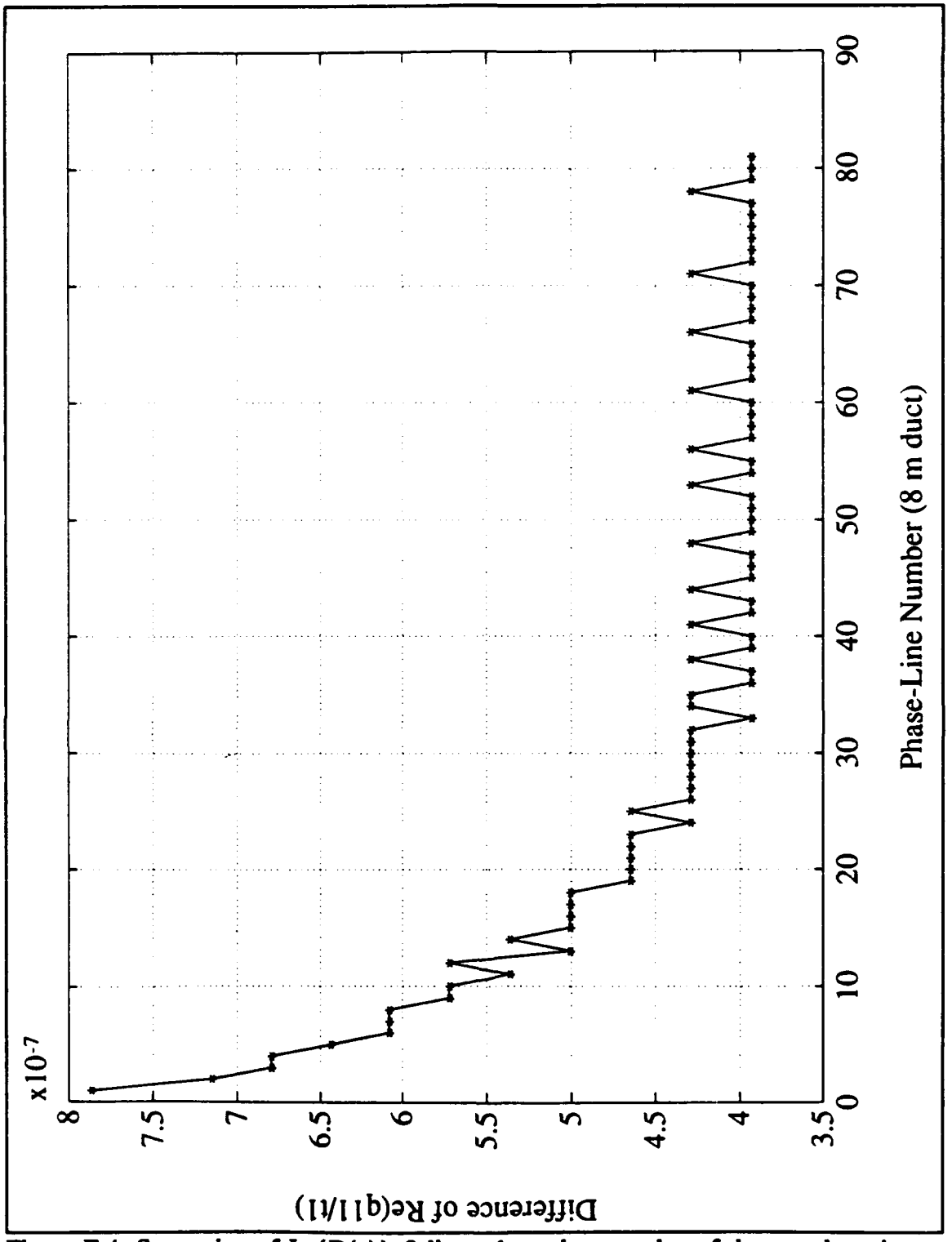


Figure E.4 Separation of Im{D(q)}=0 lines along the top edge of the search region (8 m duct).

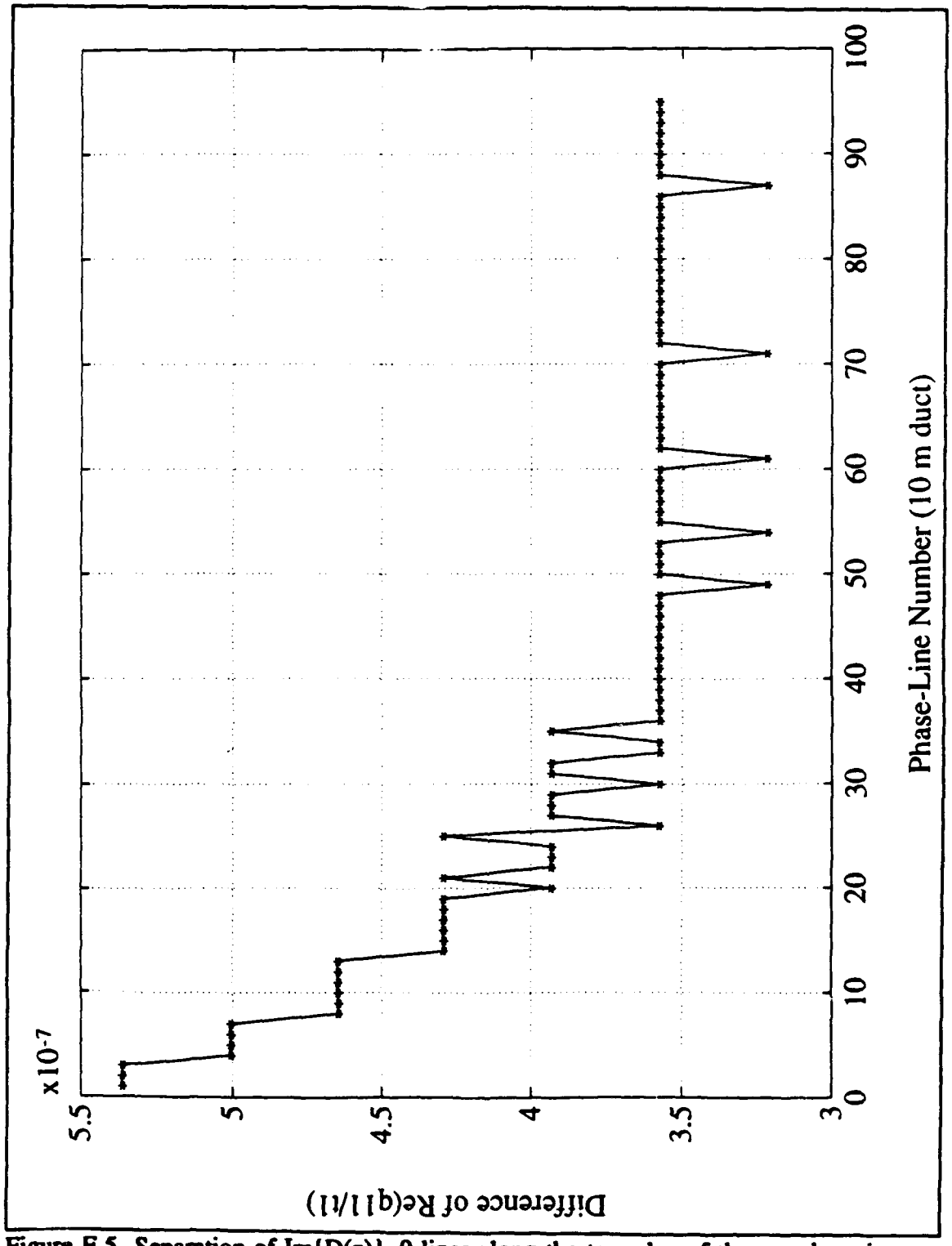


Figure E.5 Separation of Im{D(q)}=0 lines along the top edge of the search region (10 m duct).

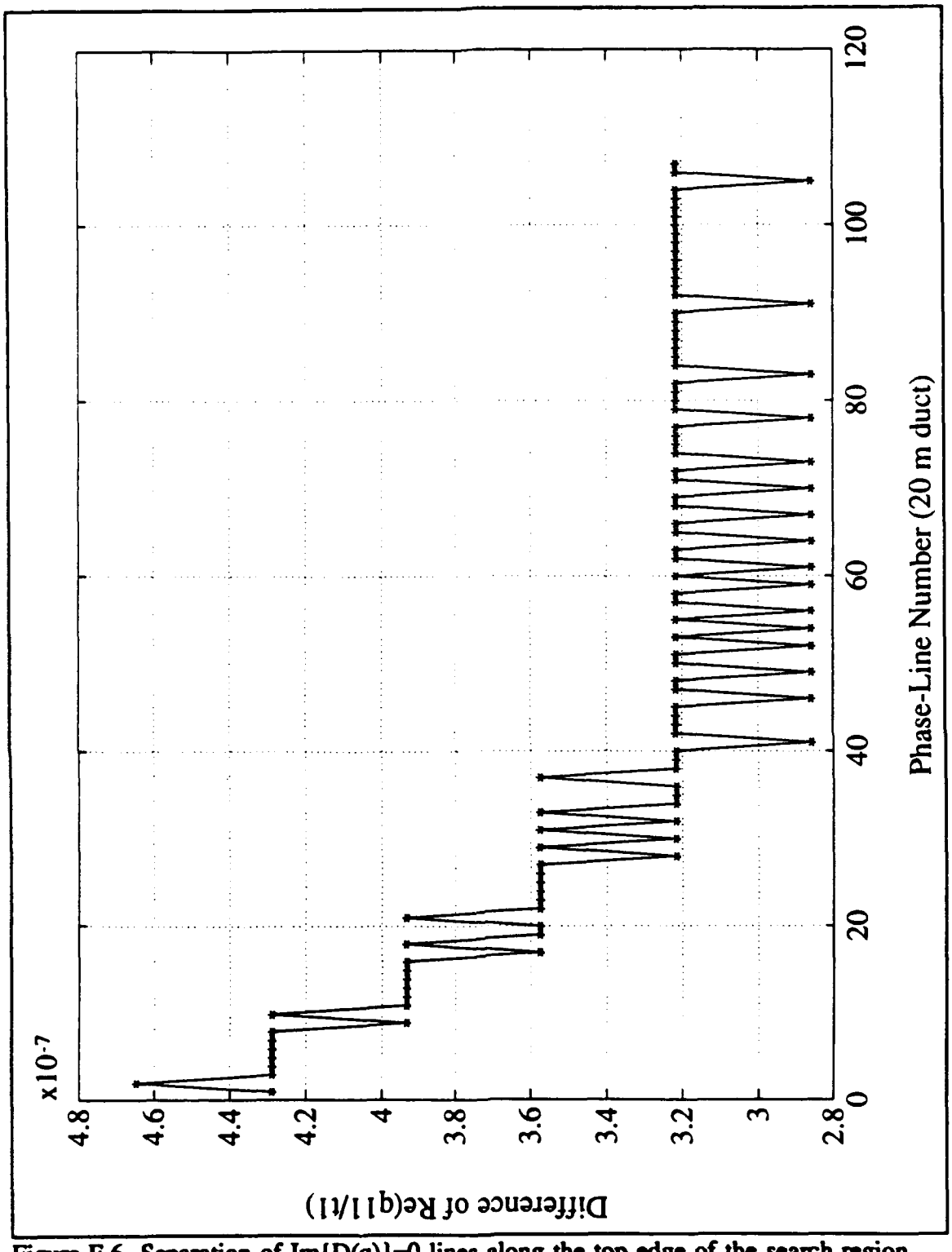


Figure E.6 Separation of Im{D(q)}=0 lines along the top edge of the search region (20 m duct).

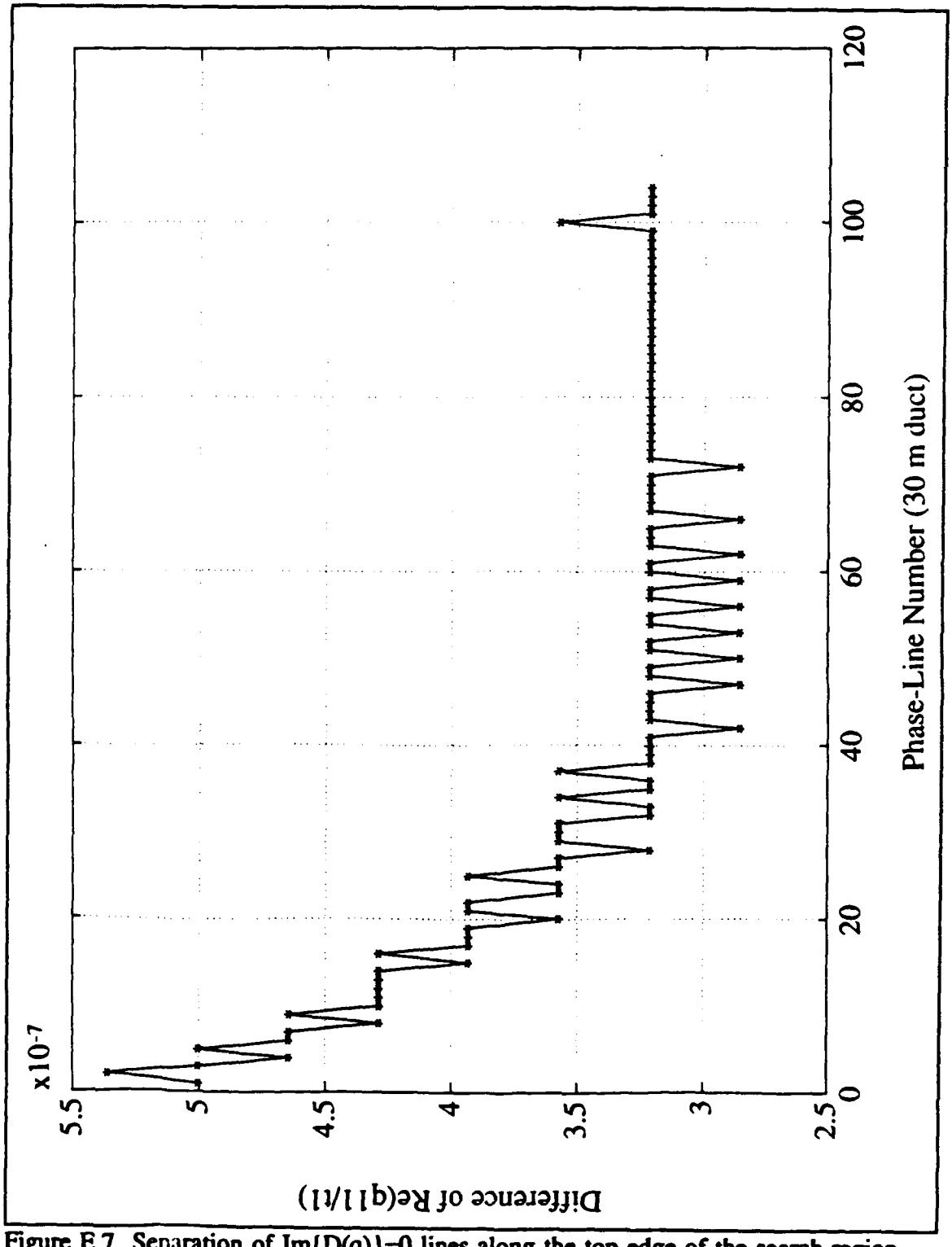


Figure E.7 Separation of Im{D(q)}=0 lines along the top edge of the search region (30 m duct).

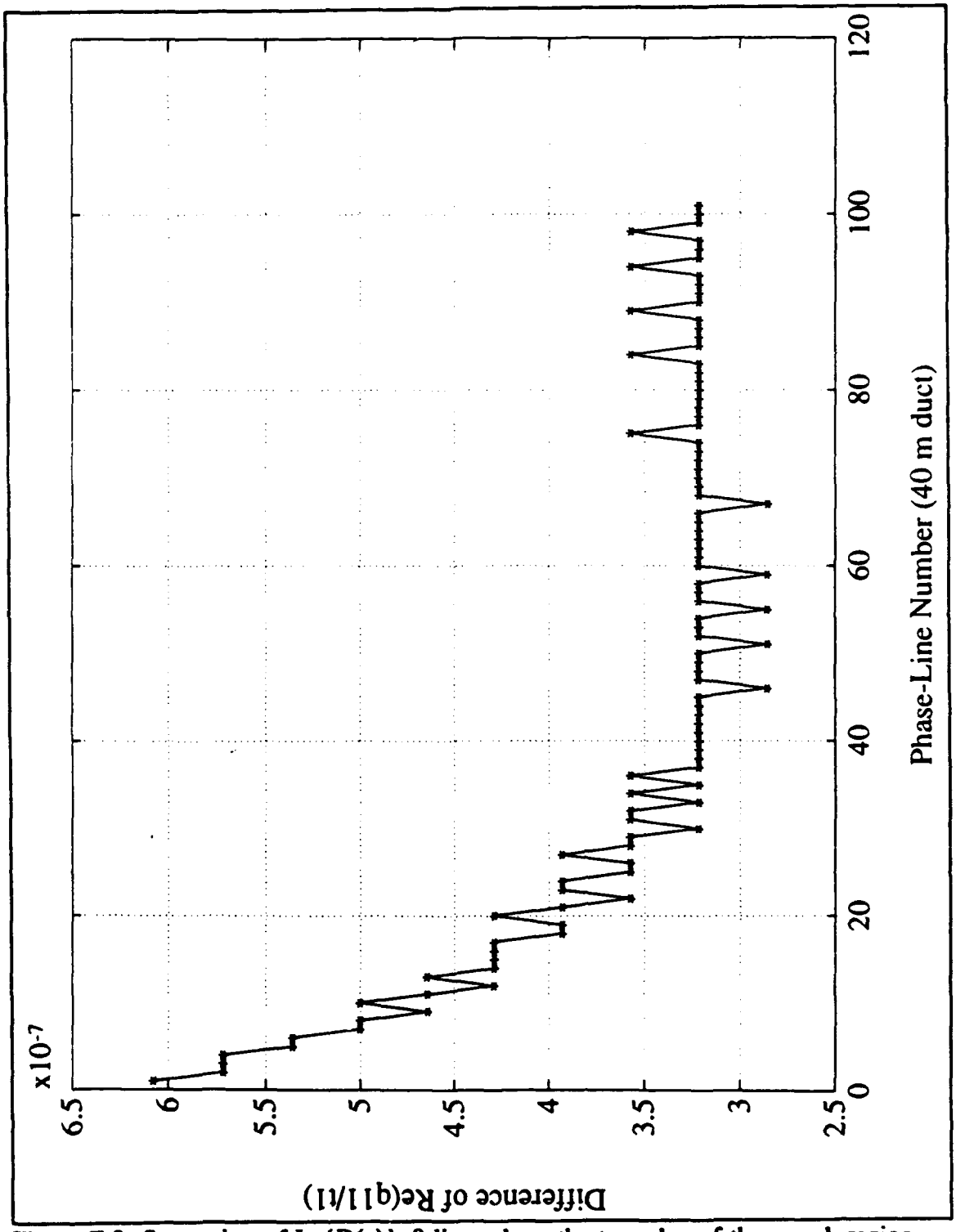


Figure E.8 Separation of Im{D(q)}=0 lines along the top edge of the search region (40 m duct).

APPENDIX F: DIFFERENCE BETWEEN CONSECUTIVE $Re(q_m/t_1)$ VALUES

Separation in real part of neighboring q-eigenvalues scaled by t_1 is plotted in this appendix against the eigenvalue locations along the real q_{11}/t_1 axis.

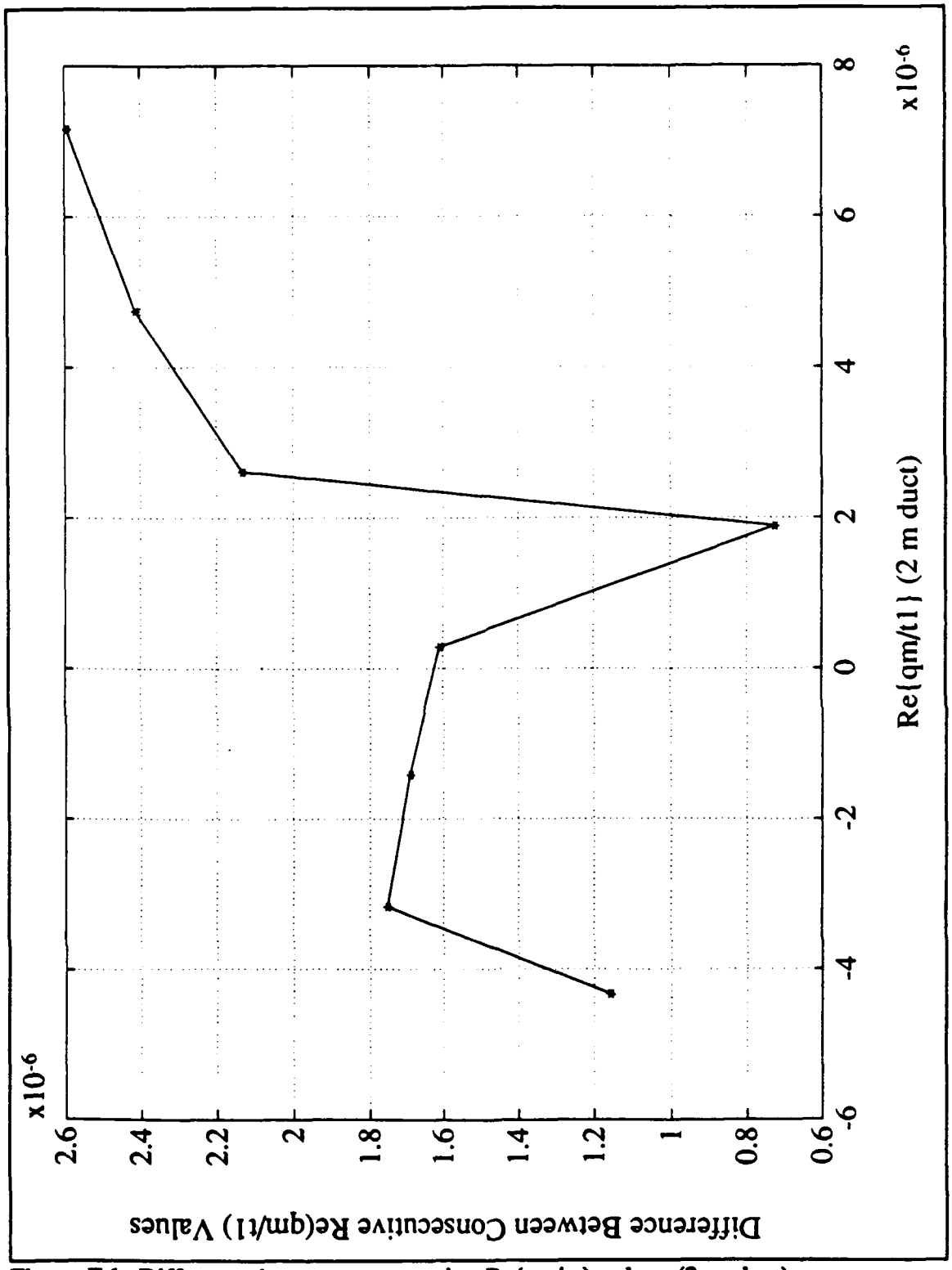


Figure F.1 Difference between consecutive $Re(q_m/t_1)$ values (2 m duct).

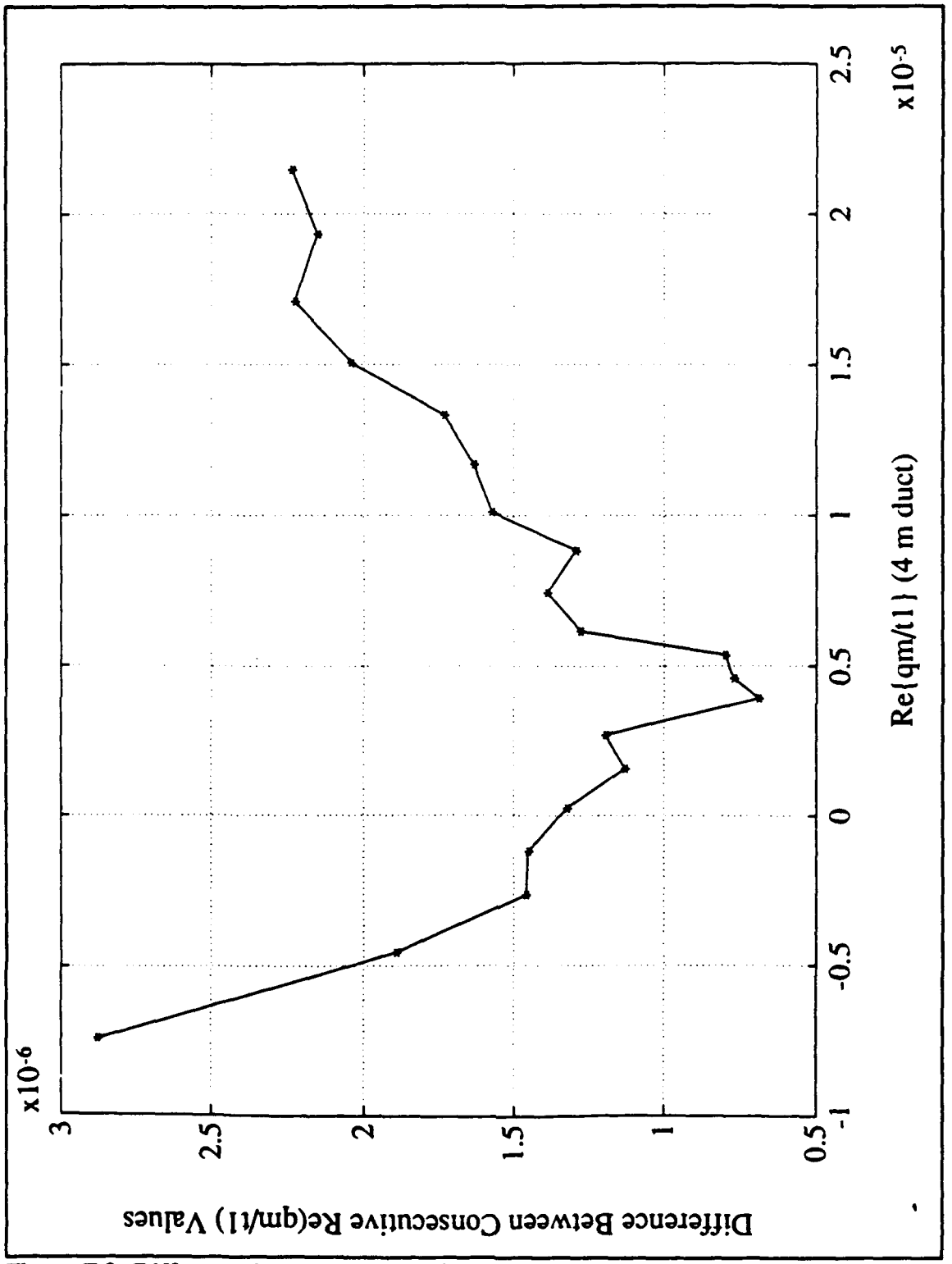


Figure F.2 Difference between consecutive $Re(q_m/t_1)$ values (4 m duct).

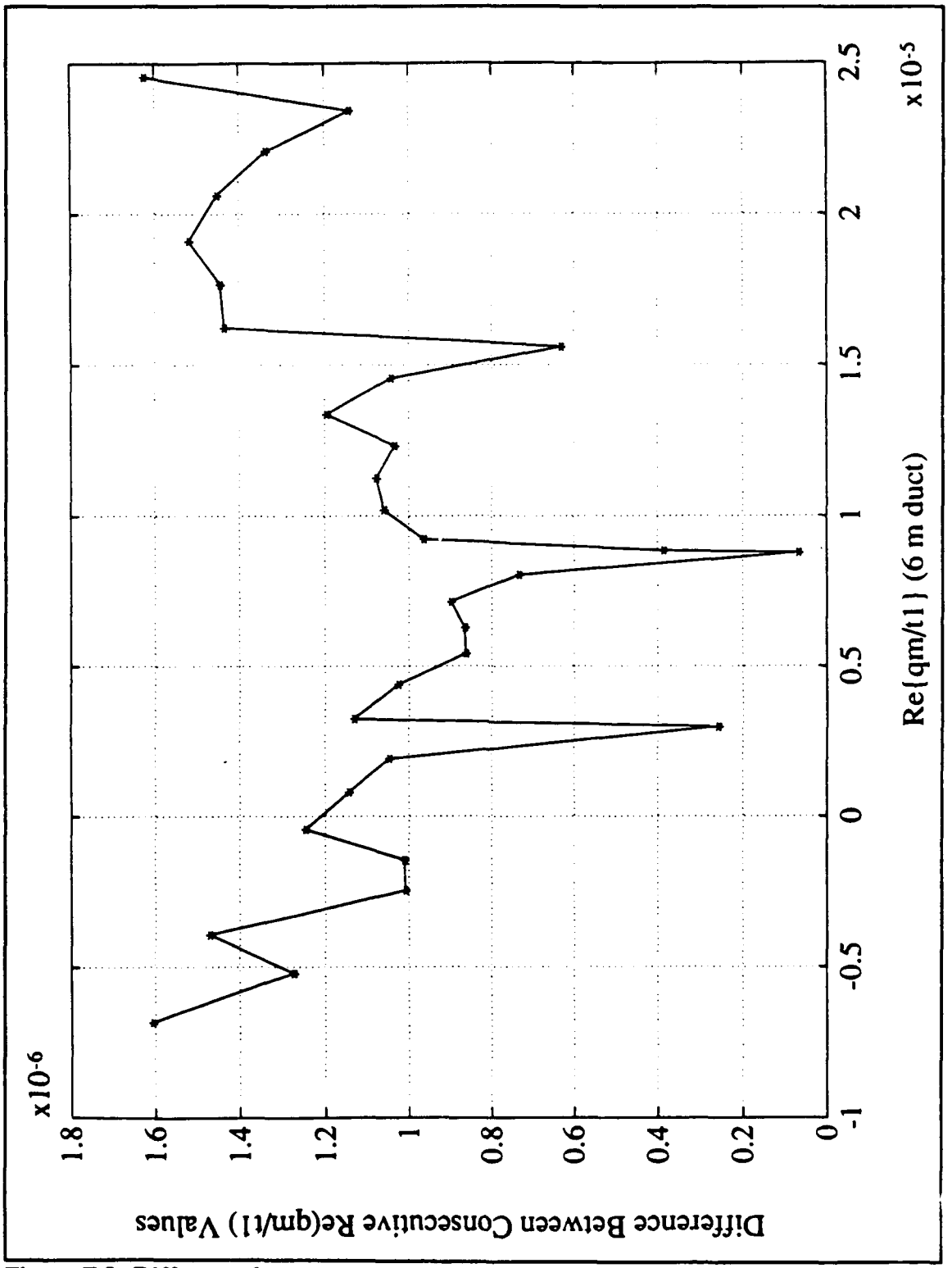


Figure F.3 Difference between consecutive $Re(q_m/t_1)$ values (6 m duct).

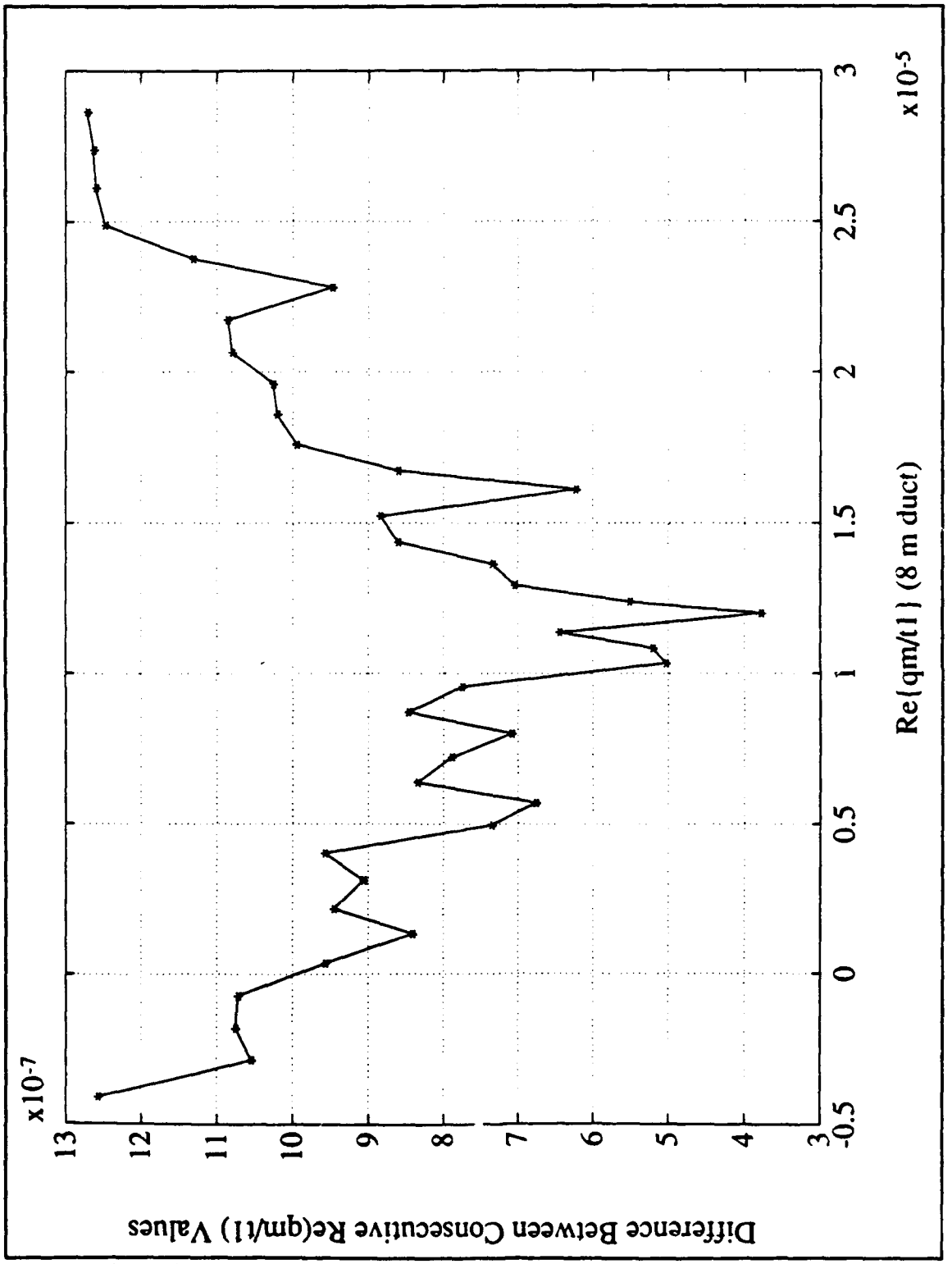


Figure F.4 Difference between consecutive $Re(q_m/t_1)$ values (8 m duct).

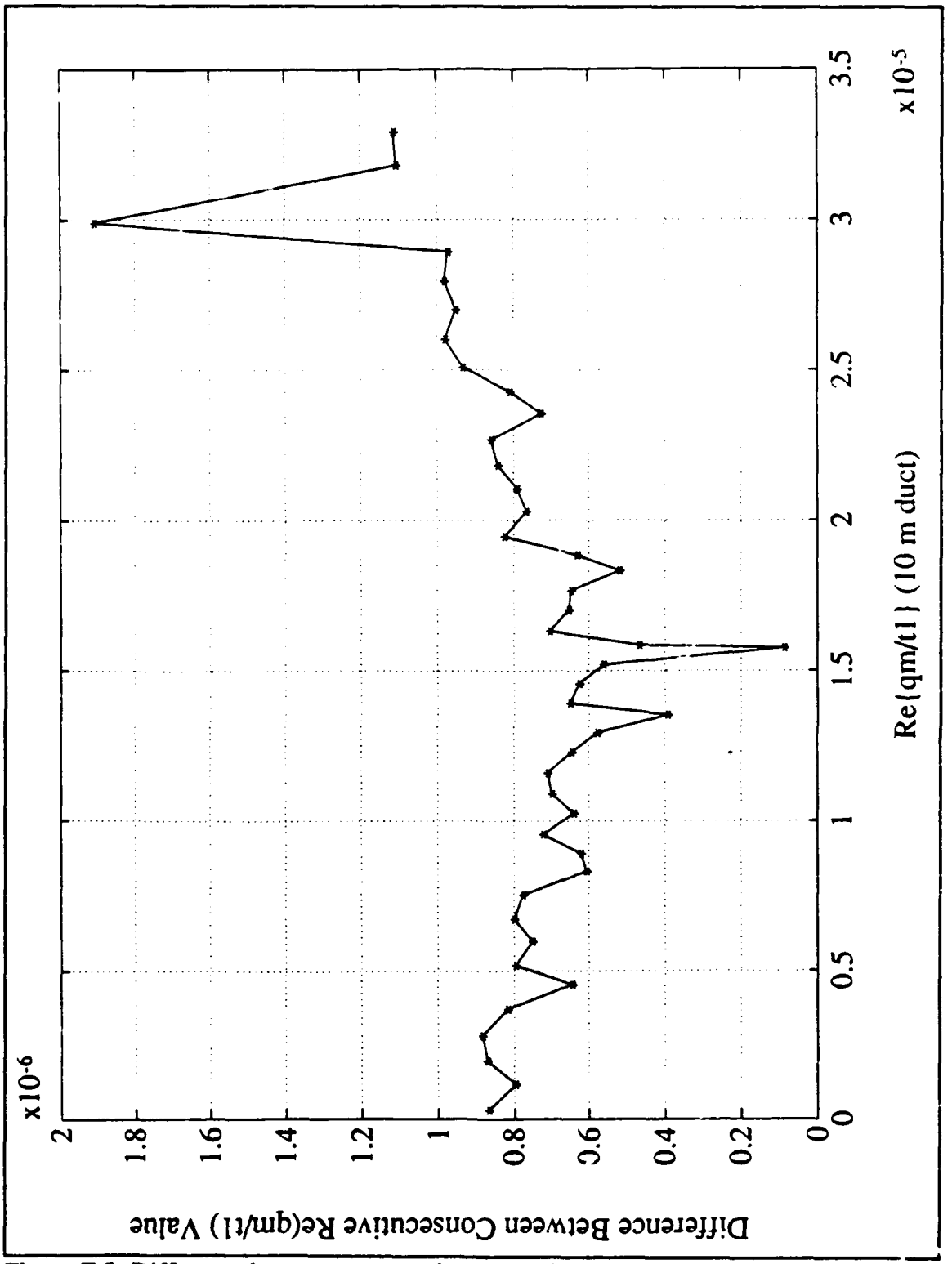


Figure F.5 Difference between consecutive $Re(q_m/t_1)$ values (10 m duct).

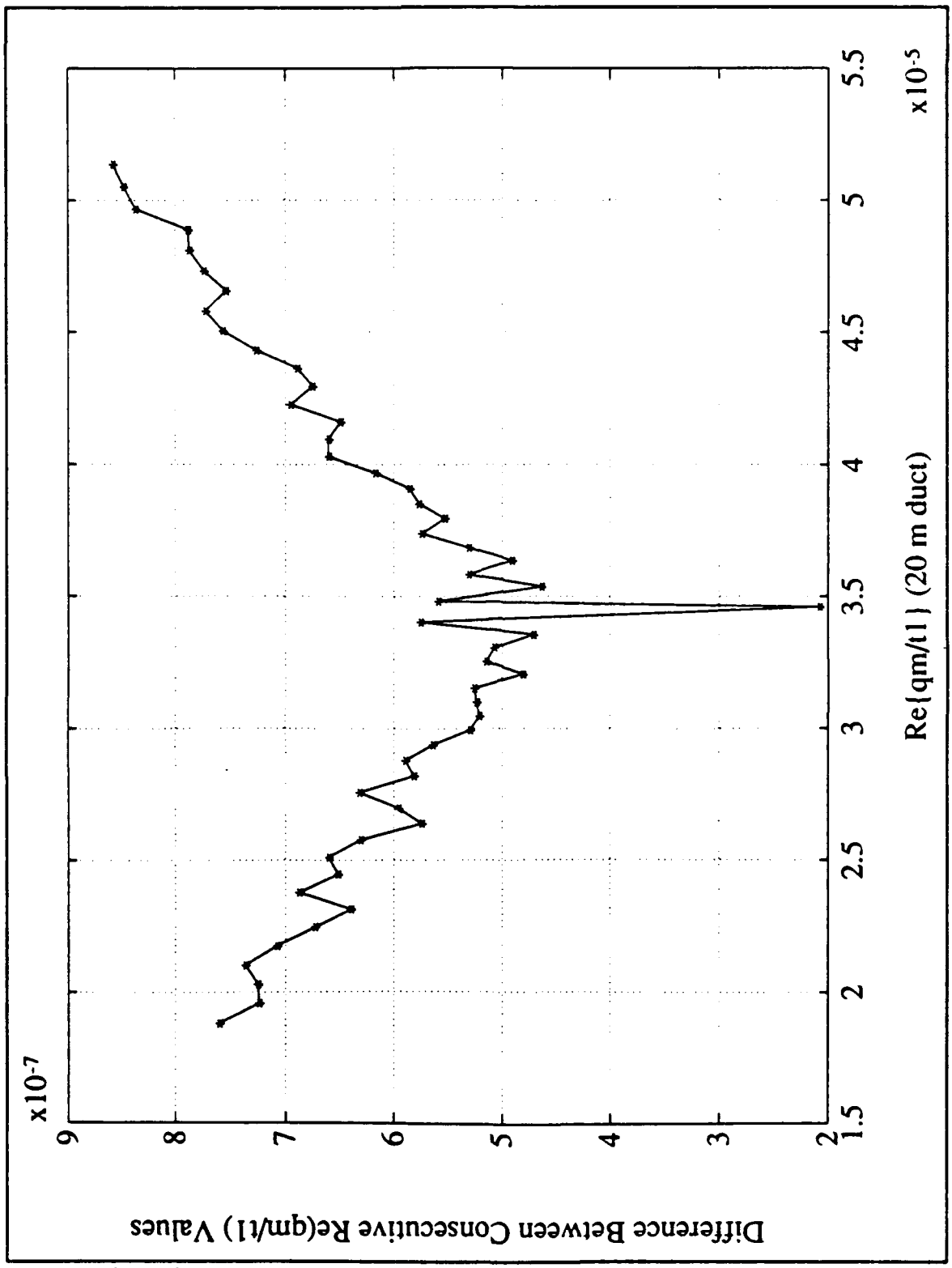


Figure F.6 Difference between consecutive $Re(q_m/t_1)$ values (20 m duct).

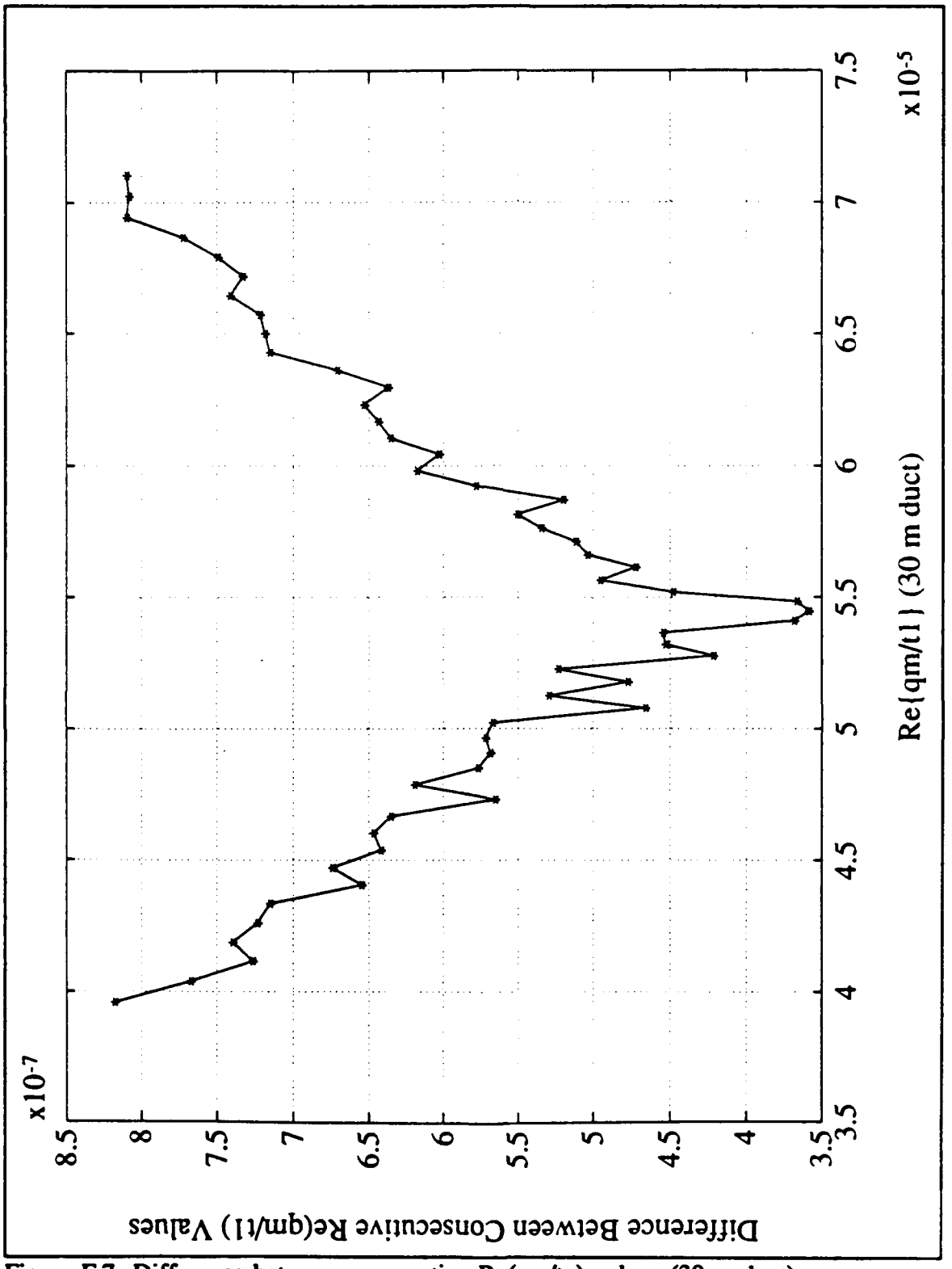


Figure F.7 Difference between consecutive $Re(q_m/t_1)$ values (30 m duct).

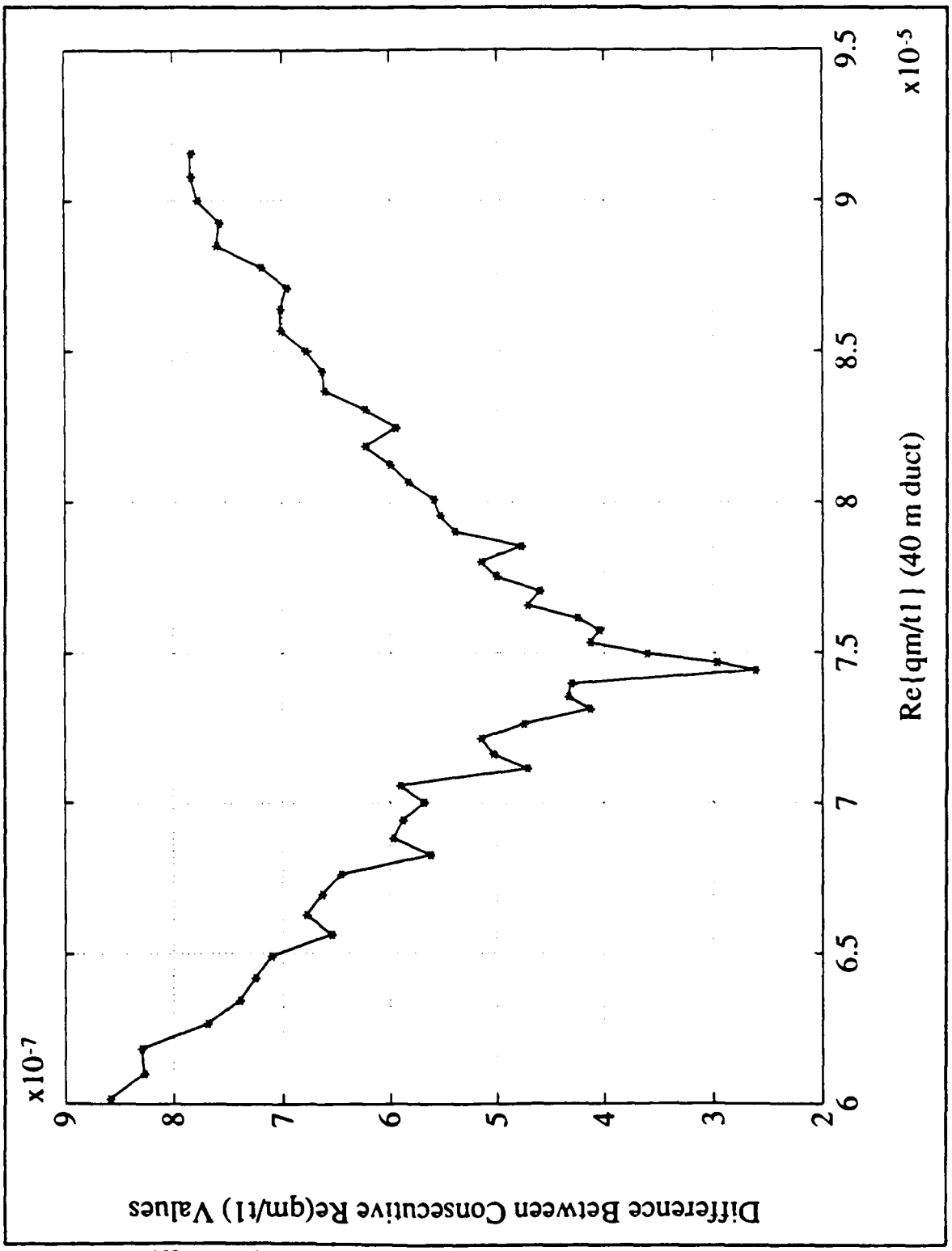


Figure F.8 Difference between consecutive $Re(q_m/t_1)$ values (40 m duct).

LIST OF REFERENCES

1. L.W. Yeoh (1990), "An analysis of M-layer: a multilayer tropospheric propagation program," *Technical Report NPS-62-90-009,* Naval Postgraduate School, Monterey, California 93943, 1990.

2. H.-M. Lee and Y.-Y. Han, "Exponential representation and consistency checking for M-Layer," *Technical Report NPSEC-92-005,* Naval Postgraduate School, 1992.

3. H.-M. Lee and Y.-Y. Han, "M-layer: NPS version," paper submitted to *IEEE Transactions on Magnetics*, special issue for the Conference on Electromagnetic Field Computation.

4. D. E. Kerr, ed., *Propagation of Short Radio Waves*, Peregrinus Ltd, London, 1987; originally published in 1951.

5. J. A. Kong, *Electromagnetic Wave Theory,* 2nd ed., John Wiley & Sons, New York, 1990.

6. K. G. Budden, *The Wave-Guide Mode Theory of Wave Propagation*, Prentice-Hall, New Jersey, 1961.

7. H.-M. Lee, "From Mie solution to Fock's theory for short range propagation," *Proceedings of the 1992 URSI International Symposium on Electromagnetic Theory*, 351-353, Sydney, Australia, August 1992.

8. S. Feferman, *The Number Systems*, 2nd ed., Chelsea Publishing Company, New York, 1989.

INITIAL DISTRIBUTION LIST

1.	Defense Technical Information Center Cameron Station Alexandria, VA 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, CA 93943-5002	2
3.	Department Chairman, Code EC Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
4.	Professor Hung-Mou Lee, Code EC/Lh Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	3
5.	Professor Lawrence J. Ziomek, Code EC/Zm Department of Electrical and Computer Engineering Naval Postgraduate School Monterey, CA 93943-5100	1
6.	Che, Jen Peng 3F, No. 6, Lane 93, Chung-Young Rd, Hsin-Tien City, Taipei County, Taiwan, R.O.C.	2
7.	Naval Academy Library Kao-Hsiung Tso-Ying P.O. Box 90175 Taiwan, R.O.C.	2
8.	Ting, Chueh Department of Electrical Engineering Chung Cheng Institute of Technology Tao-Yuan, Tai-Hsi Taiwan, R.O.C.	1